# *Orbital Gateway Interface Specification*

**V 4.2**

12-01-06

CHASE Paymentech

# Change Log

| Date | Action | Description of Change |
|---|---|---|
| 09/01/06 | Rewrite | Orbital Gateway Version 4.0 is a brand new schema that requires the specification to be rewritten |
| 11/01/06 | Updated | Added Bill Me Later in Schema version 4.1 |
| 12/01/06 | Updated | Added PINLess Debit |

# *Table of Contents*


## Introduction 8

Chase Paymentech Orbital Gateway Transaction Processing Model .....8
Transaction Types .....9
New Order: .....9
FlexCache Transactions .....10
Profile Transaction Types .....10
Mark for Capture [MFC]: .....10
Reversal [Void a Previous Transaction] .....11
End of Day .....12
Quick response .....12
Cardholder Authentication (Card Not Present) .....13
Address Verification .....13
Card Verification Numbers .....13
Verified by Visa/MasterCard SecureCode Programs .....14
Purchasing Card .....20
Introduction .....20
Edit Checks .....20
BIN Ranges .....21
Processing .....21
European Direct Debit .....23
Introduction .....23
FlexCache .....24
Transaction Types .....24
Responses .....28
Settlement .....28
Reporting .....28
Bill Me Later .....28
Introduction .....28
How it works .....29
Processing Requirements: .....29

Other ..... 30
PINLess Debit ..... 30
Introduction ..... 30
Processing Requirements ..... 31
Supported Currencies ..... 32
Virtual Terminal ..... 32
Soft Descriptors ..... 32
Introduction ..... 32
Soft Descriptor Support ..... 32
SALEM SUPPORT ..... 33
PNS SUPPORT ..... 35
Profiles ..... 35
Not a recurring service ..... 35
Benefits ..... 36
Setup Information ..... 36
Business Rules ..... 37

**Processing Interface Description 45**

Introduction ..... 45
Communication Protocol ..... 45
Posting to a URL ..... 45
Security ..... 46
Secure Sockets Layer Implementation Required ..... 46
Authentication ..... 47
XML Schema ..... 47
MIME Header ..... 48
MIME Header Field Definition ..... 48
Retry Logic ..... 51
Business Rules ..... 51
Retry Timing ..... 52
XML Request Validation on Duplicate TRACE-Numbers ..... 52
Transaction Types Supported ..... 53
Retry Error Responses ..... 53
Concurrency ..... 53
Merchant ID ..... 53

Transaction Time Out Response ... **Error! Bookmark not defined.**
Retry Attempt Time Out ... 53

**Transaction Management Messages 55**

New Order Request Elements ... 55
New Order Response Elements ... 81
Mark for Capture (of a previous authorization) Request Elements ... 85
Mark for Capture Response Elements ... 94
Void (full and partial) Request Elements ... 96
Void (full and partial) Response Elements ... 98
End of Day Request Elements ... 100
End of Day Response Elements ... 101
Profile Request Elements ... 102
Profile Response Elements ... 111
FlexCache Request Elements ... 114
FlexCache Response Elements ... 118
Quick Response Elements ... 122

**Appendix A 124**

RespCode Tag Values ... 124
AVSRespCode Tag Values ... 130
CVV2RespCode Tag Values ... 132
CAVVRespCode Tag Values ... 132
HTTP Responses ... 133
ProcStatus Tag Error Values ... 133
ProfileProcStatus Response Values ... 139
CurrencyCode and CurrencyExponent Tag Values ... 140
Purchasing Card 3 Tables ... 143
ISO Country Codes ... 143
Unit of Measure ... 144

**Appendix B 150**

General Card Validation ... 150
MOD 10 Check Digit ... 150

Card Prefix Check .....153
Card Length Check .....153

| | |
|---|---|
| Version | : 04.2 |
| Document Name | : Orbital Gateway Interface Specification V4.2 |
| Reference Documents | : *Request_PTI42.xsd* |
| | *Response_PTI42.xsd* |

© This publication is for information purposes only and its content does not represent a contract in any form. Furthermore, this publication shall not be deemed to be a warranty of any kind, either express or implied. Chase Paymentech expressly disclaims, and you expressly waive, any and all warranties, including without limitation those of merchantability and fitness for a particular purpose. Chase Paymentech reserves the right to alter product specifications without notice. No part of this publication may be reproduced or transmitted in any form or by any means, electronic or mechanical, including photocopy, recording, or any information storage or retrieval system, without Chase Paymentech's permission.

# *Introduction*

Chase Paymentech's Orbital Gateway is a proprietary XML Internet Processing System. This system works with both of Chase Paymentech's Host Processing Platforms - PNS and Salem.

Chase Paymentech maintains two proprietary Authorization and Deposit platforms. The PNS platform, which is sometimes referred to as the Tandem or Tampa, is primarily targeted to Retail and smaller customers. The Salem platform, sometimes referred to as the Stratus, is primarily targeted to Card-Not-Present and larger customers. Despite name, both systems are collocated in both Tampa, Florida and Salem, New Hampshire.

Each Platform has unique processing features, and since Orbital supports both, not all features are available to all merchants.

The Gateway processes to both Platforms using identical transaction information as presented in this specification, with the exception of any features that may only be available on one of the two Platforms. Throughout this document there will be references to BIN 000001 (Salem Platform) or BIN 000002 (PNS Platform). Please contact your technical analyst or relationship manager if you are unsure which Platform your merchant account resides on.

## Chase Paymentech Orbital Gateway Transaction Processing Model

The Chase Paymentech Orbital Gateway described in this document operates on the basis that a merchant initially instructs the gateway to perform an operation on his/her behalf. Assuming that the initial operation is successful, the gateway returns information that the merchant must use for all subsequent operations on the transaction in question.

The gateway manages the "transaction state" on behalf of the merchant. The merchant moves the transaction between the various possible states using the API messages and fields defined in this document.

# Transaction Types

The Chase Paymentech Orbital Gateway XML interface offers the following transaction types:

## New Order:

Authorization: Authorize the supplied information. This transaction type should be used for deferred billing transactions.

Authorization-Capture: Authorize the supplied information and mark it as captured for next settlement cut. This transaction should be used for immediate fulfillment.

Force Authorization-Capture: Force transactions will not generate new authorizations. A 'good' response simply indicates that the request has been properly formatted. And the Orbital Gateway will settle the captured force at the next settlement event.

Refund: Instruct the gateway to generate a refund [credit] based on the supplied information.

Complex Type Name:

- New Order Request = NewOrder
- New Order Response = NewOrderResp

### Profile Transactions in New Order [see more details in Profiles section]:

The following are the Profile actions that can be executed in a New Order Transaction

#### Using Profiles for a New Order

- One of the key transaction types is using a Profile to process a transaction.
- Overriding Profile Data: Almost any data set in the Profile can be overridden [except card type] during a transaction that is using the Profile. For instance, if a Profile included a fixed amount, but a particular transaction was for a different amount, it could be changed for that transaction by including a specific amount in the use profile request.

#### Adding Profiles during an Auth

Given that in many circumstances, the fist time a customer is setup an authorization needs to be performed, the Orbital Gateway has extended the traditional Authorization transaction to enable adding a Profile in the same request.

- Add profiles can be included with all New Order transactions types except Refunds.

## FlexCache Transactions

As opposed to using the New Order transaction type for creating new FlexCache Transactions, the FlexCache Element is used.

This supports the following FlexCache transactional capabilities:

- Card Activation:
    - Single Card Activation
    - Block Activation
    - Deactivate
    - Reactivate
- Add Value
- Authorization
- Redemption
- RedemptionCompletion
- Refund
- Balance Inquiry
- Void [Reversal]

FlexCache Reversals are generated by sending a Reversal [Void] on the original request. Please see reversal section below.

Complex Type Name:

- FlexCache Request = FlexCache
- FlexCache Response = FlexCacheResp

## Profile Transaction Types

This transaction type allows for the following profile actions [see Profiles section for details]:

- Add a Profile
- Delete a Profile
- Update a Profile
- Retrieve a profile [and all its attributes]

Complex Type Name:

- Profile Requests = Profile
- Profile Response = ProfileResp

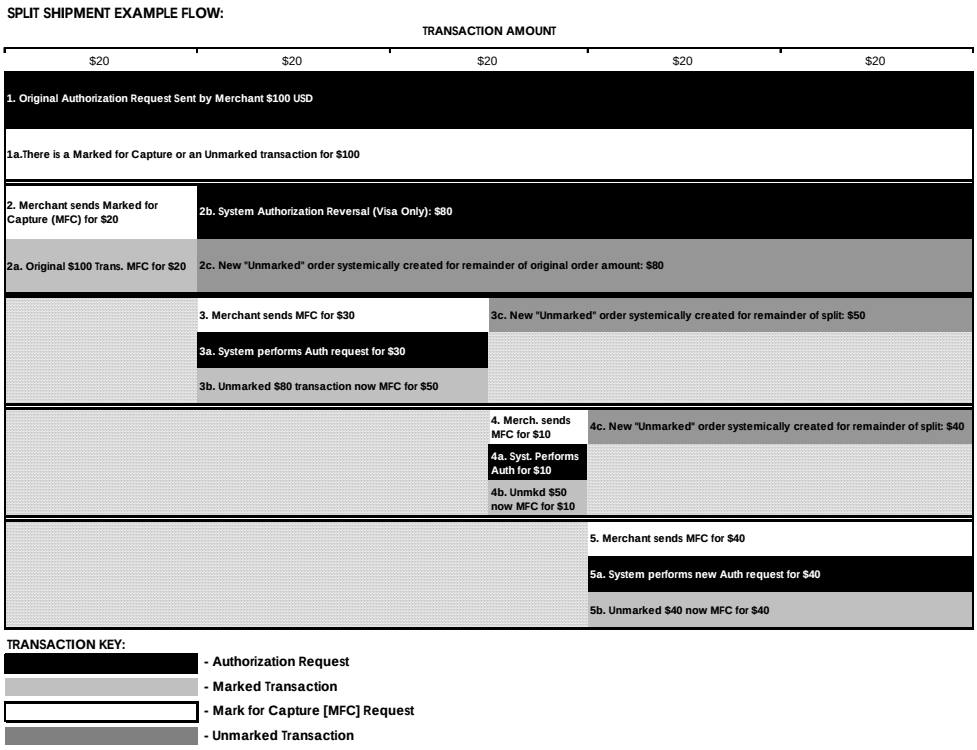
## Mark for Capture [MFC]:

Mark a previously authorized transaction as being ready to be submitted for clearing. The Mark for Capture model is present for future fulfillment models.

A transaction can be authorized now and marked for capture at any time in the next four months. The Mark for Capture can be for any amount equal to or less than the original authorization. If the amount is less than the original auth, this will be treated as a split transaction.

The split transaction also results in the creation of a new order for the balance left over from the original authorization and adjustments as required to the original authorization. Upon marking a portion or the remainder of the split transaction, the system will automatically attempt to obtain a new authorization for the new order when it is settled.

See sample split shipment scenario.


SPLIT SHIPMENT EXAMPLE FLOW:
TRANSACTION AMOUNT
$20
$20
$20
$20
$20
1. Original Authorization Request Sent by Merchant $100 USD
1a.There is a Marked for Capture or an Unmarked transaction for $100
2. Merchant sends Marked for Capture (MFC) for $20
2b. System Authorization Reversal (Visa Only): $80
2a. Original $100 Trans. MFC for $20
2c. New "Unmarked" order systemically created for remainder of original order amount: $80
3. Merchant sends MFC for $30
3c. New "Unmarked" order systemically created for remainder of split: $50
3a. System performs Auth request for $30
3b. Unmarked $80 transaction now MFC for $50
4. Merch. sends MFC for $10
4c. New "Unmarked" order systemically created for remainder of split: $40
4a. Syst. Performs Auth for $10
4b. Unmkd $50 now MFC for $10
5. Merchant sends MFC for $40
5a. System performs new Auth request for $40
5b. Unmarked $40 now MFC for $40
TRANSACTION KEY:
- Authorization Request
- Marked Transaction
- Mark for Capture [MFC] Request
- Unmarked Transaction

Complex Type Name:

- Mark for Capture Request = MarkForCapture
- Mark for Capture Response = MarkForCaptureResp

## Reversal [Void a Previous Transaction]

This transaction is for voiding a previous transaction [except FlexCache Transactions] either in the full amount or partial. If a Reversal (void) request is sent for a partial amount, the transaction will be split into two components. A voided transaction in the amount of the partial void request and the remainder of the previous transaction in the same state the full amount was previously [Authorized or Marked for Capture].

Notes:

- A Reversal/Void transaction does not reversal the original Authorization for any card type other than FlexCache.
- Transactions can not be voided once they have settled. So Captured Authorizations and Refunds must be voided before the end of day action [whether auto-settle or customer triggered].

Complex Type Name:

- Void Request = Reversal
- Void Response = ReversalResp

## End of Day

An "End of Day" request/response instructs the gateway to submit all transactions previously marked for capture [including all successful refunds] for clearing.

Alternative End of Day methodologies include:

- Auto-Settle: At a Merchant ID level, an account can be setup to settle automatically at any given 15 minute increment during the day and in any US based time zone.
- Virtual Terminal: End of Days can be triggered using the Orbital Virtual as many times as desired. Please see Virtual Terminal Users Manual for instructions.

Complex Type Name:

- End of Day Request = EndOfDay
- End of Day Response = EndOfDayResp

## Quick response

When a transaction has an error condition, such as a time out condition or a poorly formed message request, the gateway will generate a quick error message back to the requestor. This error response takes the form of a "Quick Response".

Complex Type Name:

- Quick Response = QuickResp

# Cardholder Authentication (Card Not Present)

## Address Verification

Address Verification, also known as AVS, is a cardholder authentication mechanism available to merchants. In addition to providing merchants with an additional risk management tool, it is required by Visa to qualify for the lowest interchange rates and protects against certain chargeback conditions. As such, it is highly recommended by Chase Paymentech that all transactions include this information. Two keys on AVS are:

- The minimum required data for AVS is the Cardholders billing ZIP.
- AVS is only supported by credit cards issued in the United States, Canada and the United Kingdom.
- For PNS routed accounts [BIN 000002], the Orbital Gateway does not support ZIP formats outside the US Convention of NNNNN and NNNNN-NNNN. If a transaction is submitted with a ZIP outside this data convention, the entire transaction will reject for invalid ZIP structure.
- For Salem routed accounts [BIN 000001], the Orbital Gateway will accept ZIPs formatted alpha-numeric with a length between 1 and 10 in length. These ZIP's will be forwarded to the Salem authorization host.
- The Card Types that support AVS are:
    - Visa
    - MasterCard
    - American Express
    - Discover

## Card Verification Numbers

The Orbital Gateway supports the submission of Card Verification Numbers [see more detailed definition below] for which the methods of payments it is available. The specific field name for this value in the XML interface is <CardSecVal>.

### Visa CVV2/MasterCard CVC2/Discover CID Programs

The Orbital Gateway supports Visa's CVV2 (Card Verification Value 2), MasterCard's CVC2 (Card Validation Code 2) and Discover's CID fraud reduction programs. This section will provide some background information on supporting these programs.

The value for these cards is 3 digits and can be found on the signature panel on the reverse side of the credit card and is represented by the three digits following the account number.

This value may not be stored at all. Not even for future transactions, as it is against regulations to do so.

The use of this value provides an important security check due to the fact that only the individual in possession of the actual credit card will be able to provide the value to the merchant. Statistics validate those individuals who may know the account number, but are not in possession of the actual credit card perpetrating much of the fraud occurring in the non-face-to-face environment.

When a merchant collects this value and passes it in the authorization request, Chase Paymentech passes this data through the authorization system to the card issuer. In the authorization response, the card issuer validates the accuracy of CVV2/CVC2/Discover CID value for the specific card. Used in conjunction with the valid expiration date, this service provides a valuable tool for assessing the true cardholder has placed the order with you for your services or product.

### American Express CID Merchant Processing Requirements

American Express provides a similar program to Visa, MasterCard and Discover except with a few differences:

- The value for these cards is 4 digits and is printed, not embossed, on the front of all cards. On the American Express card it appears on the right border of the card, however on Optima cards, it appears on the left border of the card.
- In situations where the CID value is invalid, American Express could respond with an authorization decline message. American Express will respond with a separate CID response code.
- To process American Express CID, the merchant must contact their American Express Client Manager to have their American Express service establishment numbers flagged to accept the Amex CID value. American Express will not provide validation of the CID value if the merchant's Service Establishment (SE #) is not enabled.

### FlexCache/Stored Value Application

The Chase Paymentech FlexCache™ program supports CVD2 (Card Verification Data 2), which is also known as a PIN, as an optional feature determined by the merchant. The four-digit value may be imprinted on the back of the stored value card, and used to facilitate a secure Card Not Present transaction when the consumer wishes to use a FlexCache card as their method of payment.

### Guidelines for populating Card Security Fields

The Card Security Fields are called <CardSecValInd> & <CardSecVal>.

## Verified by Visa/MasterCard SecureCode Programs

Verified by Visa and MasterCard SecureCode are both solutions designed to authenticate cardholders when paying online. These products offer a mechanism for securing the Internet channel by strongly authenticating the cardholder at the point of interaction by providing a unique transaction-specific token that provides evidence that the cardholder originated the transaction.

### How it works

*Verified by Visa*
Verified by Visa is based on the 3-D Secure Protocol, which uses Secure Sockets Layer (SSL) encryption to collect and protect payment card information transmitted via the Internet. It uses three domains for the authentication process:

- Issuer Domain – Where the Issuer is responsible for determining whether authentication is available for the card account presented in a purchase.
- Acquirer Domain – Where the Acquirer accepts Internet transaction data from the Merchant and passes it to Visa.
- Interoperability / Visa Domain – This is operated by Visa, where transaction information is exchanged and stored using 3-D Secure as the common protocol.

Transaction Flow
The cardholder shops at participating Internet Merchants with no changes to the shopping or checkout. The cardholder selects the merchandise to be purchased and proceeds to the checkout. At the checkout, the cardholder may complete the purchase and payment information a variety of ways, including self-entered, an electronic wallet; Merchant one-click, or using other checkout capabilities.

After the purchase and payment information is entered, the cardholder selects the "buy" button. This activates the Merchant Server Plug-In (MPI) software application, which checks its local cache to determine if the Visa Issuer BIN participates in Verified by Visa. If the BIN is participating, the MPI will generate an inquiry to the Visa Directory Server to determine if the cardholder's account is enrolled in Verified by Visa. The Visa Directory Server sends a Verify Enrollment Request message to the Issuer Access Control Server (ACS) to determine if authentication is available for the cardholder's account number. The Visa Directory Server sends the Issuer ACS response to the MPI. If authentication is not available, the Merchant server receives an "authentication not available" message and returns the transaction to the Merchant's commerce server to proceed with a standard Authorization request. If authentication is available, the message response provides the URL for the Issuer ACS where the cardholder can be authenticated. The MPI sends a message and script directing the cardholder's browser to establish a pop up session with the Issuer ACS.

The browser directs the transaction to the URL specified for the Issuer ACS creating an SSL session. The Issuer ACS displays a inline web page to the cardholder. The page includes Issuer-specific and Visa branding, transaction details (including Merchant name and sale amount), and prompts the cardholder to enter their password. The cardholder is allowed a limited number of password attempts, typically 3 to 5, as defined by the Issuer ACS. If unable to correctly enter the password, the cardholder may access the password hint that was established during the registration. If the password is entered correctly, the transaction continues. If the cardholder is not registered, the ACS briefly displays a processing window and the transaction continues as an attempted authentication. If the password is incorrectly entered more times that the Issuer limit, the failed Payer Authentication Response is returned to the Merchant.

The Issuer ACS retrieves the authentication information and compares it against the data that was registered during the initial registration process. If the data matches, a success page is presented to the cardholder and the Issuer ACS sends a message through the browser to the merchant, thus providing evidence of cardholder authentication. Using the Issuer's encryption keys and transaction data, the Issuer server calculates the Cardholder Authentication Verification Value (CAVV), which will be included with the Electronic Commerce Indicator (ECI), as provided at the time of authentication by the MPI, in the response to the Merchant.

The Issuer ACS creates, digitally signs, and sends a Payer Authentication Response to the cardholders browser, and sends transaction information to the Visa Authentication History Server (AHS) for storage. All Payer Authentication Response messages – successful, unable to authenticate, failed, and attempted authentications – are transmitted and stored in the AHS. The browser routes the Payer Authentication Response back to the MPI, which validates the digital signature in the response, verifying that it is from a valid participating Issuer. If the digital signature is verified and the Issuer has sent an approved Payer Authentication Response, the cardholder is deemed authenticated and the MPI returns the transaction to the storefront software. The Merchant starts processing the order, determining whether it can be fulfilled, and calculating taxes and shipping for the total transaction amount.

The Merchant will send the CAVV, ECI [<authenticationECIInd>] of 5 (authenticated transaction) or 6 (attempted authentication) to the Orbital Gateway. The CAVV must be sent in Base 64 encoding within the XML Document. If the CAVV is not submitted in Base 64 encoding or if the CAVV is sent with a non-eCommerce transaction, a response code of 37 in the XML element <RespCode> will be returned.

Chase Paymentech will pass the CAVV and ECI along to Visa with the authorization request. These fields are used during authorization processing to verify that authentication, or attempted authentication, was performed and to qualify for the eCommerce Customer Payment Services.

The Issuer receives the authorization request, validates the CAVV and responds with a CAVV Response Code [or <CAVVRespCode> within the XML response document] as well as an approval or a decline of the authorization. If the CAVV does not match, the Issuer should decline the transaction.

Visa has not implemented any new decline codes for Verified by Visa. The standard decline codes should apply.

**NOTES: A Merchant may not submit for authorization a purchase transaction that has failed authentication.**

*MasterCard SecureCode*

- MasterCard SecureCode is a solution designed to authenticate cardholders when paying online. SecureCode offers a mechanism for securing the Internet channel by strongly authenticating the cardholder at the point of interaction by providing a unique transaction-specific token that provides

evidence that the cardholder originated the transaction. SecureCode uses MasterCard's Universal Cardholder Authentication Field (UCAF) infrastructure to communicate the authentication information among the cardholder, Issuer, merchant and Acquirer.

- MasterCard SecureCode supports the 3-D Secure Protocol (same as Verified by Visa). MasterCard SecureCode requires merchants to install a 3-D Secure v1.0.2 compliant Merchant Server Plug-in (MPI) software.

### Transaction Flow

The cardholder shops at a participating SecureCode Internet Merchant with no changes to the shopping or checkout. The cardholder selects the merchandise to be purchased and proceeds to the checkout. At the checkout, the cardholder may complete the purchase and payment information in a variety of ways, including self-entered, an electronic wallet; Merchant one-click, or using other checkout capabilities.

After the purchase and payment information is entered, the cardholder selects the "buy" button. The customer shopping experience is the same for both of the Issuer platforms up until the time that the Merchant Order Confirmation page is displayed.

The MPI activates and checks its local cache and the MC Directory Server to determine if the customer card number is part of a participating MasterCard SecureCode BIN range. If so, a Verify Enrollment Request message will be sent from the MPI, to the MC Directory Server and forwarded to the Issuer Access Control Server (ACS) to determine if authentication is available for the cardholders account number. The MC Directory Server sends the ACS response to the MPI. If authentication is available, the message response provides the web address for the Issuer ACS where the cardholder can be authenticated. If authentication is not available, the Merchant server receives an "authentication not available" message and returns the transaction to the Merchant's commerce server to proceed with a standard Authorization Request. There is no "attempted SecureCode" transaction type unlike Verified by Visa [ECI = 6].

The MPI sends a message and script directing the cardholder's browser to establish an inline web page session with the Issuer ACS. The window displays Issuer-specific and MasterCard branding, transaction details – including Merchant name and amount, and prompts the cardholder to enter their SecureCode (e.g. password). If the password is entered correctly, the transaction continues. The cardholder is allowed a limited number of password attempts, typically 3 to 5, as defined by the Issuer ACS. If unable to correctly enter the password, the cardholder may access the password hint that was established during the registration. If the password is incorrectly entered more times than the Issuer limit, a failed Payer Authentication Response is returned to the Merchant.

The Issuer ACS retrieves the authentication information and compares it against the data that was registered during the initial cardholder registration process. If the data matches, a success page is presented to the cardholder and the Issuer ACS sends a message through the browser to the Merchant providing evidence of cardholder authentication, including a 28-byte Account AAV. This AAV is

generated cryptographically using Issuer-specific secret keys that are synchronized with keys at the Issuer's authorization platform.

*Merchant Submission of AAV Data*

The merchant will then send the transaction to Chase Paymentech, along with the 28-byte AAV in Base 64 encoding, within the Orbital Gateway XML Interface Specification.

If the AAV is not submitted in Base 64 encoding or if the AAV is sent with a non-eCommerce transaction, a response code of 37 in the XML element <RespCode> will be returned. Also, if the Merchant has not tested and certified with Chase Paymentech to participate in MasterCard SecureCode, and an AAV is sent with the e-Commerce transaction, a response code of 38 in the XML element <RespCode> will be returned, which indicates the Merchant should contact their Chase Paymentech Representative to become SecureCode enabled.

Chase Paymentech will forward the transaction, including the AAV in the MC authorization request. The Issuer receives the authorization request, validates the AAV and responds with an approval or a decline of the authorization. If the AAV does not match, the Issuer should decline the transaction.

MasterCard has not implemented any new decline codes for SecureCode. Standard decline codes apply.

### Merchant Requirements

Merchant Plug-in Software

Install a Certified 3-D Secure Merchant Plug-in Software Application or code to the 3-D Secure Protocol.

Verify that Merchant Plug-in will provide the CAVV and or AAV in Base 64 encoding before sending to Chase Paymentech. If not, Merchant will need to convert to Base 64 before sending to Chase Paymentech.

### Business Rules

There are a number of business rules in terms of when a CAVV and or AAV should be presented on aged transaction, reauthorizations, split transactions, etc. The Orbital Gateway abstracts your interface from many of these issues. This section will outline what these rules are and what is necessary to understand from an Interface Perspective.

*Authorizations*

Merchants are required to request authorization for all Verified by Visa and MasterCard SecureCode eCommerce transactions. Merchants must supply the CAVV and ECI on all Visa authorization attempts and the AAV on all MasterCard Authorization attempts.

*Failed Authorizations*

Merchants are prohibited from submitting transactions for authorization that have failed authentication.

*Late Fulfillment*

When a participating merchant splits the shipment of an order, each authorization component may be submitted with the authentication data (ECI of

5 or 6 [6 is for VbV only] and the CAVV or AAV) of the original purchase. In the event of a dispute, the Acquirer must be able to establish that the authorization requests were related to a single customer authenticated purchase. Furthermore, if a Deposit record is sent for the subsequent shipment, the authorization will already have been tagged as "used"; therefore, in order to receive the full benefit of Verified by Visa and MC SecureCode, a merchant must send the authentication data with the subsequent deposit record so that when Chase Paymentech reauthorizes, the authentication data can be sent as well.

MasterCard SecureCode

Initial SecureCode authorization requests with AAV's older than 30 calendar days may be declined by the Issuer.

*Recurring Transactions*

When a participating Merchant offers services of an ongoing nature to a cardholder for whom the cardholder pays on a recurring basis (for example, insurance premiums, subscriptions, Internet service provider fees, membership fees, tuition, or utility charges), the cardholder payments are considered recurring payments.

If the first payment originated as an Electronic Commerce Transaction via the Internet, it must be submitted with the appropriate Electronic Commerce Indicator (ECI) value, including Verified by Visa or MasterCard Secure Code authentication data (CAVV or AAV respectively), if applicable.

All subsequent payments must be submitted as Recurring transactions [see Orbital Gateway Interface Specification and Message Templates documents). The merchant must not store and submit the CAVV with any subsequent transaction.

*Currencies Supported*

All Currencies

### Chargeback Liability Sift Exclusions

*Verified by Visa*

There are certain exclusions from the Chargeback provisions related to attempted authentications. They are:

1. All Visa Commercial Cards (Visa Business, Visa Purchasing and Visa Corporate Cards), anonymous Prepaid Cards (such as gift cards), and transactions from new channels (such s mobile devices) are excluded from chargeback protections for attempted authentications. If these cards are enrolled in Verified by Visa and the Issuer authenticates the cardholder, the Issuer is not permitted to submit a chargeback for unauthorized usage disputes (reason codes 23, 61, 75, and 83). Either the Issuer ACS or Visa may designate excluded transactions; however, the Visa Directory Server will over-ride excluded responses from an Issuer ACS if the BINs are not also designated as excluded BINs in the Visa Directory. The designation of BINs as Commercial or anonymous Prepaid Cards must be consistent with VisaNet.
2. Transactions conducted in new channels (such as mobile or wireless devices).

Merchants named in the Global Merchant Chargeback Monitoring Program are not eligible for Chargeback protection for attempted authentications during the time that they are required to participate in the program and three months thereafter. Visa will work with Acquirers to ensure compliance with this requirement. There are no additional steps for Issuers regarding this provision.

# Purchasing Card

## Introduction

The Orbital Gateway supports the processing of procurement cards by fully supporting the enhanced data required by Visa and MasterCard for both Level 2 and Level 3 data. Additionally, for American Express, the Orbital Gateway for Salem customers supports Level 2 and enhanced TAA.

Purchasing Cards with Level 3 data are typically used in a business-to-business environment providing and collecting funds for outstanding invoices. Merchants have the ability to collect their funds in conjunction with the settlement of their credit card transactions and still provide their customer with the necessary line item detail; thus, providing a cleaner process for both the merchant and their customer.

## Edit Checks

The Orbital Gateway performs edit checks on incoming data to ensure necessary information is present. In the event necessary information is missing from a transaction, the transaction will result in an error [see list of error codes in the ProcStatus listing in Appendix A]. Data fields that are edited by Chase Paymentech have been marked as "Conditionally Required" in the Transaction Management section of this document. Additionally, there some special edit checks specific to each host described below.

### PNS

There are two key mathematical data validations specific to PNS processing for Purchasing Card 3 Processing.

- The amount field [<PC3Dtllinetot>] of every line item must equal the Unit Cost [<PC3DtlUOM>] multiplied by the quantity [<PC3DtlQty>] less any discounts [<PC3DtlDisc>]. If it does not, then this transaction will receive an error.
- Additionally, the sum of all the Line Item totals [<PC3Dtllinetot>] cannot exceed the transaction amount [<Amount>] submitted for an order.

### Salem

There is no mathematical validation for Purchasing Card 2 or 3 for Salem customers.

## BIN Ranges

The BIN range assigned by the card associations can identify purchasing cards. BIN ranges are subject to change at the discretion of the card associations.

## Processing

Purchasing Card 2 or 3 data can be sent either in the original auth [via an Auth or Auth-Capture] or it can be appended to the authorization via the Mark for Capture request if it was not originally supplied in the authorization request.

There are different rules for adding and adjusting Purchasing Card data via the Mark for Capture based on whether it is simply level 2 data or if it is level 3 data.

### MFC Adjustment of Level 2 Data

When processing Purchasing Card level 2 data, the additional data can be included in either the Auth (and adjusted) or added altogether via the Mark for capture transaction.

The following scenarios describe the optional behavior:

- If the Purchasing data is submitted on the Authorization.
    - No purchasing Card data is submitted on a Mark for Capture [whether full or partial amount]. The gateway will submit at settlement the purchasing card data presented on the authorization.
    - Purchasing Card Data is submitted on the Mark for Capture [MFC] and the MFC is for:
        - The full amount: The Purchasing Card data submitted on the MFC will override the data submitted on the Auth.
        - Partial Amount: Where the amount of the MFC is less than the auth and a split transaction is generated, whatever data is submitted on the first mark for capture will be used on all splits. If each split should have different data, then each MFC should include the relevant purchasing card data. But that is not required.
- If the data is not submitted on the auth, then it must be included on the MFC for it to be submitted at Settlement.
    - Where the amount submitted on the MFC is equal to the Auth, the transactions is complete and that data will be used.

- Where the amount is less, just as described above, and a split transaction is generated, whatever data is submitted on the first mark for capture will be used on all splits.

### Purchasing Card 3

Just as with Purchasing Card level 2, when processing level 3 data the additional data can be included in either the Auth (and adjusted) or added altogether via the Mark for capture transaction. However, because of the PNS based Purchasing Card Validation rules [see edits above], there is different behavior in terms of what can be done when adjusting the purchasing data via a Mark for Capture.

The following scenarios describe the optional behavior:

- If the Purchasing data is submitted on the Authorization.
    - No purchasing Card data is submitted on a Mark for Capture [whether full or partial amount]. The gateway will submit at settlement the purchasing card data presented on the authorization.
    - Purchasing Card Data is submitted on the Mark for Capture [MFC] and the MFC is for:
        - The full amount: The Purchasing Card data submitted on the MFC will override the data submitted on the Auth, as long as the amended data is still consistent with the data validation rules [PNS/Tampa customers only] otherwise the MFC capture request will fail.
        - Partial Amount: Where the amount of the MFC is less than the auth and a split transaction is generated, whatever data is submitted on the first mark for capture will be used on all splits. If each split should have different data, then each MFC should include the relevant purchasing card data. But that is not required, as long as the amended data is still consistent with the data validation rules [PNS/Tampa customers only] other wise the MFC will fail.
- If the data is not submitted on the auth, then it must be included on the MFC for it to be submitted at Settlement.
    - Where the amount submitted on the MFC is equal to the Auth, the transactions is complete and that data will be used. Again, for PNS/Tampa customers, the purchasing card level 3 data must pass the data validation rules or the MFC request will fail.
    - Where the amount is less, just as described above, and a split transaction is generated, whatever data is submitted on the first mark for capture will only be used for that transaction [for Salem and PNS customers alike]. All subsequent MFC requests [again both Salem and PNS] must include the purchasing card level 3 data relevant to that component [and for PNS/Tampa customers it must additionally match the amount, based on the edits, of the MFC.]

### Virtual Terminal

All of the functionality supported through this interface for Purchase Card 2 and 3 is additionally available through the Orbital Gateway Virtual Terminal.

# European Direct Debit

## Introduction

### Overview

European Direct Debit [EU DD] is a popular method of payment for merchants marketing in Europe. While any merchant may want to accept direct debit payments, it is most important and cost effective for those merchants collecting recurring payments. Unlike in the US, many EU customers prefer to pay for recurring services by direct debit to their bank accounts. This is especially true in Germany where almost 40% of all electronic payments are made by direct debit.

### How it works

In Europe, each country operates its own direct debit network. Merchants wishing to accept direct debit throughout Europe would face the requirement to establish banking relationships and technical integration for each country in which they wish to market. Chase Paymentech Solutions has created a single technical interface for direct debit processing for multiple countries.

### Processing Requirements:

Merchants must contract with Chase Paymentech Solutions for acceptance of European Direct Debit. The Merchant Descriptor is defined on the vendor's system. Sending the Merchant Descriptor record will not alter the descriptor on the accountholder's statement.

The purpose of this document is intended to outline how a developer can code to take advantage of this method of payment within the Orbital Gateway, both in terms of the message layout and the business rules.

### Virtual Terminal

All of the functionality supported through this interface for European Direct Debit is additionally available through the Orbital Gateway Virtual Terminal.

### Platforms

The Orbital Gateway only supports the European Direct Debit method of payment through the Salem host platform [BIN 000001]. This method of payment is not supported on the PNS host [BIN 00002].

# FlexCache

The Orbital Gateway supports Chase Paymentech's proprietary Stored Value Card product called FlexCache™ for both Salem and PNS customers. See the FlexCache Message Specification for more detail.

## Transaction Types

This section defines all the FlexCache transaction types supported by the Orbital Gateway.

### Card Activation Transaction Types

The following is a list and description of the Card Activation transaction types.

*Activate*
This transaction is used to activate one individual card for the first time.

Merchants processing to the PNS Host can process Prior Activation Transactions by additionally passing the correct prior approval code. If the valid Prior Approval code is not passed, it will be treated as a new Activation request.

Salem Merchants attempting to process a Prior Activation will receive an error response.

*Block Activate*
Block activation provides for the ability to activate more than one card at a time. The maximum number of cards that can be activated at a time is 100. In a Block Activate request, the card number of the first card in a series is defined, plus the number of additional sequential cards.

If a Block Activation fails, none of the cards in the block will be activated. And the first card number that caused the Block Activation failure will be returned in the response.

The Virtual Terminal supports the ability to perform a Block Activation of 10,000 in a single request. However, as indicted above the On-Line interface maximum is only 100 cards per request.

*Deactivate*
This transaction is for the deactivation of a live card. Passing an amount is not required for this transaction type.

*Reactivate*
There are two mechanisms for reactivating a card once it has been deactivated:

- Reversing the deactivation transaction which will return the card to the same balance prior to the deactivation transaction.
- Or the card can be reactivated. In a reactivation transaction, a dollar amount must be passed for how much the card for which will be reactivated.

NOTE: The Orbital Gateway supports $0 activation transactions for PNS [BIN 000002].

### Add Value

This transaction type adds value to an active card. If an Add Value is performed on an inactive card, it will both activate the card and perform the add value action as well.

Merchants processing to the PNS Host can process Prior Add Value Transactions by additionally passing the correct prior approval code. If the valid Prior Approval code is not passed, it will be treated as a new Add Value request.

Salem Merchants attempting to process a Prior Add Value will receive an error response.

### Purchase and Refund Transactions

The following transaction types are for purchases and refunds. There are two different transaction combinations available for purchases.

1. Authorization followed by a Redemption Completion – This transaction combination is only valid for Salem based customers.
2. Redemption.

These two combinations allow for different purchase processing behavior on FlexCache cards. This section will define how each transaction type functions.

*Authorization*

Almost all FlexCache transaction types immediately affect the card balance, meaning add or reduce the funds based on the result. In some circumstances there might be a desire to perform a sale wherein an authorization is performed and the funds not actually moved. One reason for this, for example, might be a deferred shipment of goods.

The Authorization transaction does exactly that. It will reduce the "Available to Buy" amount without actually reducing the actual funds.

Once the item has been shipped, performing a Redemption Completion can complete the transaction, see below.

Generally speaking, an authorization will hold the requested funds for seven ("7") days, after which the funds will be available again.

As stated above, this functionality is only available Merchants processing through the Salem Platform.

There are two different optional behaviors when managing Redemption Completions. The next two sections outline those options.

*Partial Redemption*

FlexCache supports a functionality called Partial Redemption. If for whatever reason, when the Redemption Completion is submitted, the amount of the original authorization exceeds the available balance, the merchant has two options on how to treat this transaction, which is managed by submitting the XML Element <FlexPartialRedemptionInd>.

If the available balance on the card is less than the Redemption Completion Amount:

- The transaction can be declined with no amount redeemed from the card. If this is the desired behavior on a particular transaction, do not submit this element or null fill it.
- Or the transaction can be approved, with the maximum amount of the Redemption completion fulfilled, even though it is less. The response in this circumstance would include both the Requested amount and the actual redeemed amount. The behavior can be implemented by passing the <FlexPartialRedemptionInd> element with a value of "Y".

### *Auto-Authorization of the Remainder*

As stated above, when a Redemption Completion is submitted for an amount that is less than original amount, the entire original balance on hold is removed and the new amount redeemed from the card. If the transaction was a split shipment; however, the merchant might want the card to maintain a hold on the balance for the remaining un-redeemed amount. To re-establish this hold on the funds, a new authorization needs to be submitted.

- This new authorization can either be a whole new request submitted by the merchant.
    - Or, when performing the Redemption Completion, the Element <FlexAutoAuthInd> can be passed in the Mark For Capture request with a value of Y. When this happens, if the initial Redemption Completion is successful, the Orbital Gateway will automatically generate a new Authorization Request for the amount that represents the un-redeemed amount from the initial Authorization.
    - Return a single response identifying the result of the Redemption Completion and, if it was successful, the new authorization request. If there is a new authorization, the following aspects of the response are important to understand:
        - There will only be one set of Current and Previous Card Balances returned and those values will be reflective of the result of both actions.
        - If the new authorization request is successful, the new authorization will be a distinct transaction with a distinct Transaction Reference Number from the original Authorization. All the relevant details of the second request will be returned in the response.

### *Redemption Completion*

As stated above, a redemption completion is to complete an authorization. A Transaction Reference Number [<TxRefNum>] is returned which references the original transaction. Assuming the authorization approved, a then a Redemption Completion FlexAction is submitted and additionally including the

original auths transaction reference number and the amount to be settled [this amount can be equal to or less than the original authorization. When an amount is less than original amount, the entire original balance on hold is removed and the new amount redeemed from the card.]

As stated above, this functionality is only available Merchants processing through the Salem Platform.

*Redemption*
As opposed to an Authorization followed by a Redemption Completion, a Redemption transaction is the mechanism to perform an immediate redemption. Once completed, Redemptions can only be reversed.

Merchants processing to the PNS Host can process Prior Redemption Transactions by additionally passing the correct prior approval code. If the valid Prior Approval code is not passed, it will be treated as a new Redemption request.

Salem Merchants attempting to process a Prior Redemption will receive an error response.

*Refund*
This transaction type is for initiating refunds to a FlexCache card. It is essentially the same as an Add Value transaction.

### Reversals

All transaction types, excluding Balance Inquiries, can be reversed, thus returning a transaction to the state it was in prior to the action being reversed. There are two restrictions as it relates to processing Reversals:

- For Salem customers, the reversal must be performed within seven ["7"] days of the original transaction.
- For PNS based customers, the reversal must be performed before the next batch close. Batch closes for FlexCache are usually automatically at 5:00am EST, regardless of what the Auto-settle time is on the Gateway.
- For all customers, another action has not occurred that no longer makes the reversal possible.
    - For example, an active card can no longer have an activation reversed once a transaction has been processed. The card can only be deactivated at that point, if desired.

A reversal is accomplished by simply processing a 'Void' FlexAction type using the merchant information and the Transaction reference number of the original. This is true of all reversal transaction types. Unlike other transactions, the Complex Type 'Reversal' is not used for FlexCache Void/reversal actions.

The response on a Reversal will provide the same information as any other response [Current Balance, Previous Balance, Response Codes, etc]. In addition, it will identify specifically what transaction type is being reversed, such as "Reversal – Redemption" in the <FlexAction> tag. See section 0 for more information on responses.

#### Balance Inquiry

This transaction will perform a Balance Inquiry.

### Responses

The basic authorization response for all FlexCache transactions is the same. In other words, all responses will be returned in the same basic XML Document form with the same base minimum data elements. The transactions types that will include more information are:

- Block Activations [if they fail]
- Redemption Completions with the Partial Redemption Flag
- Redemption Completions with Auto-Authorization behavior.

### Settlement

Since transactions affect the balance of a card real time, FlexCache transactions are not affected by the End of Day process options. Instead transactions will automatically fall into one of two buckets when viewed through the virtual terminal:

- Open FlexCache items: This will include all un-settled activity:
    - Authorizations that have not been redeemed [Redemption Completion]
    - Declined transactions
    - Errors
- In the Review section of the Virtual Terminal, all Redeemed items will be viewable. These items will be grouped on a daily basis on the same timing as the FlexCache System reports activity, which is 5am – 5am.

### Reporting

All true FlexCache reporting is available from the FlexCache system, including Re$ource On Line. Any questions about the available reports should be directed to your account manager.

The Virtual Terminal should not be used for FlexCache reconciliation.

## Bill Me Later

### Introduction

Bill Me Later is an innovative and secure payment solution for card-not-present merchants. The Bill Me Later method of payment is a non-plastic issued credit vehicle that manages the consumer payment function by providing a

transactional credit decision in lieu of the standard predetermined credit line and associated authorization process. Bill Me Later allows consumers to make online/mail order purchases without inputting credit card information.

## How it works

Using proprietary credit scoring and fraud detection capabilities, I4Commerce screens each Bill Me Later transaction in real time, instantly decisioning all Bill Me Later requests made by customers.

## Processing Requirements:

Merchants must contract with I4Commerce for acceptance of Bill Me Later.

The Orbital Gateway enforces the following data requirements for sale [authorization, authorization-capture] transaction types:

- Required:
  - Account Number
  - Bill To Address [see standard AVS elements]
  - Ship To Address [see AVSDest elements]
  - Shipping Cost [BMLShippingCost]
  - Terms and Conditions Version [BMLTNCVersion]
  - Customer Registration Date [BMLCustomerRegistrationDate]
  - Customer Type Flag [BMLCustomerTypeFlag]
  - Item Category [BMLItemCategory]
  - Customer Birth Date [BMLCustomerBirthDate]
  - Customer Social Security Number [BMLCustomerSSN]
  - Product Delivery Method [BMLProductDeliveryType]
- Optional:
  - Customer Source IP [BMLCustomerIP]
  - Customer Email [BMLCustomerEmail]
  - Pre-approval Invitation Number [BMLPreapprovalInvitationNum]
  - Promotional Code [BMLMerchantPromotionalCode]
  - Customer Annual Income [BMLCustomerAnnualIncome]
  - Customer Resident Status [BMLCustomerResidenceStatus]
  - Customer Checking Account [BMLCustomerCheckingAccount]
  - Customer Saving Account [BMLCustomerSavingsAccount]

NOTE: Please contact your I4Commerce Bill Me Later Integration Analyst during the requirements definition phase prior to development to determine required fields.

#### Currencies

US Dollar Only

### Other

#### Virtual Terminal

All of the functionality supported through this interface for Bill Me Later is additionally available through the Orbital Gateway Virtual Terminal.

#### Platforms

The Orbital Gateway only supports the Bill Me Later method of payment through the Salem host platform [BIN 000001]. This method of payment is not supported on the PNS host [BIN 00002].

## PINLess Debit

Customers use their ATM/Debit cards to pay for goods or services rather than cash, check, or credit card.

Debit transactions are always authorized on a "real time" basis with the actual authorization resulting in the debit of the customer's bank account. These transactions must still be captured and settled to Chase Paymentech to support funding, reporting and associated reconciliation.

The Orbital Gateway presently offers PINLess Debit Processing as an option for Salem [BIN 000001] customers.

### Introduction

It is more commonly known as Debit Bill Payment.

This is a debit transaction where neither the magnetic stripe contents nor the PIN is part of the authorization message.

PINLess Debit is currently only supported by the three largest debit networks: Star, NYCE, and Pulse.

The debit network rules for PINLess debit programs are strict and the networks that support these transactions must approve the merchant prior to their accepting PINLess debit transactions. As a result, PINLess debit processing is only available to merchants in selected industries, specifically utilities, telephone companies, cable TV providers, some insurance companies, government entities, and financial institutions. This list could change so check with your account manager for availability rules.

The customer must initiate these transactions via the Web or IVR and the merchant assumes 100% liability for these payments.

Please refer to the *Debit Bill Payment User Manual* for card association and debit network regulations.

## Processing Requirements

As a result of the specific processing rules associated with PINLess Debit, the Orbital Gateway enforces specific behavior as it relates to PINLess Debit:

- Only allow Authorization-Capture transaction types are allowed. This means:
  - No Auth Only [future fulfillment] transactions.
  - No Mark for Capture
  - No Splits
  - No voids.
  - No refunds.
- All Merchant ID's [Transaction Divisions] enabled for PINLess Debit must have auto-settle enabled.
- PINLess Debit BIN Ranges are every dynamic. Chase Paymentech makes PINLess Debit BIN files available on a regular basis to determine which cards are eligible to participate on each transaction. Additionally, the Orbital Gateway imports and stores the most up to date PINLess Card ranges. If a card is submitted as PINLess Debit [as identified by the required card mnemonic] and it is not in an eligible card range, a ProcStatus error code of 9717 ["PINLess Debit: Card Number Not Eligible for PINLess Debit Processing"]. Optionally, a merchant can retrieve their own Debit BIN files. Please discuss with your account manager the available options.
- A new Industry type of IVR [IV] has been added but is only allowed PINLess Debit method of payment.
- Data set required to submit a PINLess Debit transaction is:
  - IndustryType – IVR or eCommerce Only
  - MessageType – AC only
  - BIN – 000001 only
  - Merchant ID
  - TerminalID – 001 only
  - CardBrand – DP
  - Account Number
  - Expiration Date - null
  - Currency Code – 840 [US Dollar] only
  - Currency Exponent – 2 only
  - Order ID
  - Amount
  - Biller Reference Number
- Approved PINless Debit transactions may return a Blank or N/A authorization code.

### Supported Currencies

U.S. Currency

### Virtual Terminal

The Orbital Virtual will display and report PINLess debit transactions, but it will not support:

- Creation of new PINless debit transactions are not supported because PINless Debit transactions are only supported via IVR or ECommerce channels.
- Adjustments of existing PINLess debit transactions.

# Soft Descriptors

### Introduction

The Soft Descriptor Records are used to define the merchant name/product description that will appear on the consumer's statement. It allows the merchant greater flexibility in describing the consumer's purchase.

It is subject to issuer discretion whether this descriptor will be displayed on the cardholder statement.

### Soft Descriptor Support

Support for Soft Descriptors is not globally available to all customers using the Orbital Gateway:

- Salem [BIN 000001] - The Orbital Gateway supports Soft Descriptors into the Salem Host. However:
    - Prior Risk Department approval is required.
    - The Merchant ID/Terminal ID must be enabled for Soft Descriptors on the Orbital Gateway.
    - Please contact you Chase Paymentech Representative.
- PNS [BIN 000002] - The Orbital Gateway supports Soft Descriptors into the PNS Host. However:

- It is only supported for Chase Paymentech Canada customers.
- The Merchant ID/Terminal ID must be enabled for Soft Descriptors on the Orbital Gateway.
- The behavior is different from that of the Salem Interface. Please refer to the PNS development section to understand how it works.
- Please contact you Chase Paymentech Representative.

## SALEM SUPPORT

### Rules and Guidelines – Credit Card

The description in the *merchant name field* should be what is most recognizable to the cardholder. It should consist of the company name and /or trade name combined with some type of description of the product or service that was purchased.

Chase Paymentech will not generate or segregate reports by the Soft Descriptor. If the Merchant wishes to see Salem reports segregated by product, the Merchant must set up specific reporting divisions and deposit those transactions under that division number.

For those Merchants who need to roll up several merchant names under one corporation, please contact your Chase Paymentech Representative for details on the use and regulation of the Soft Descriptors.

- The Merchant Name can be one of 3 different Lengths:
  - 3 bytes
  - 7 bytes
  - 12 bytes
- An addition Product Description can be appended based on the length of the Merchant Name, such that they are a combined length of 21 bytes. In other words, the options are:
  - 18 bytes
  - 14 bytes
  - 9 bytes

*Notes:*

- The City Field allows the merchant to identify the business location or provides the cardholder with a Customer Service Phone Number or URL. This is a requirement to qualify for Visa's lowest Direct Marketing interchange rate.
- If the merchant submits a backslash (\) in the merchant descriptor, then it will be converted into a hyphen (-) on the cardholder statement. If the merchant submits a question mark (?) in the merchant descriptor, then it will be converted into a space on the cardholder statement.

- There are certain American Express card types/programs that ignore the descriptors send using Soft Descriptors. The Optima card is one of these types. The merchant should contact their American Express representative for more details.
- Non-eCommerce sent with a URL will not qualify for the best interchange.
- For MasterCard MOTO (Transaction Type 1) and Recurring (Transaction Type 2), if the City/Phone field at the division level is not a Customer Service Phone Number, then a Customer Service Phone Number must be populated in the Merchant city/Customer Phone Number field or the transaction will reject with Response Reason Code BP ("Customer Service Phone reqd. on Tran Types 1(MO/TO) and 2(Recurring). MC Only").
- The Orbital Gateway will apply the asterisks ['*'] in the necessary locations. Please do not add these to the request.

### Rules and Guidelines – ECP

The Automated Clearing House (ACH) uses two fields to describe the transaction to the consumer. The Merchant Name, 15 bytes, will always appear on the consumer's statement, and the Entry Description, 10 bytes, will appear on the consumer's statement a majority of the time. Both are required fields.

Chase Paymentech recommends that the Merchant Name be used for the Doing Business As (DBA) description and the Company Entry field be used for the product description.

When utilizing the Soft Descriptor for ECP transactions, both the Merchant Name and the Entry Description are mandatory.

### Soft Descriptor Examples:

A couple of different examples of Soft Descriptors are:

- 3 Byte Merchant Descriptor with Phone #

```
<SDMerchantName>XYZ</SDMerchantName>
<SDProductDescription>PAYMENT1OF3</SDProductDescription>
<SDMerchantCity/>
<SDMerchantPhone>888-888-8888</SDMerchantPhone>
<SDMerchantURL/>
<SDMerchantEmail/>
```

- 12 Byte Merchant Descriptor with Email

```
<SDMerchantName>XYZCOMPANY</SDMerchantName>
<SDProductDescription>PYMT1OF3</SDProductDescription>
<SDMerchantCity/>
<SDMerchantPhone/>
<SDMerchantURL/>
<SDMerchantEmail>suppt@xyz.com</SDMerchantEmail>
```

- ECP

<SDMerchantName>XYZCOMPANY12345</SDMerchantName>
<SDProductDescription>PRODUCT123</SDProductDescription>
<SDMerchantCity/>
<SDMerchantPhone/>
<SDMerchantURL/>
<SDMerchantEmail/>

## PNS SUPPORT

### Rules and Guidelines

Again, the support for Soft Descriptors via the PNS Host is only for customers processing through Chase Paymentech Canada.

Unlike Salem, the only value that gets passed on the Cardholder statement is the Merchant Name field. And for these customers, it is a maximum of 25 bytes of data.

All other Soft Descriptor fields can optionally be sent, but will not be submitted to the settlement host and will not display on the cardholder statement.

### Soft Descriptor Examples:

```
<SoftDescriptor>
        <SDMerchantName>XYZ PAYMENT1OF3</SDMerchantName>
        <SDProductDescription/>
        <SDMerchantCity/>
        <SDMerchantPhone/>
        <SDMerchantURL/>
        <SDMerchantEmail/>
</SoftDescriptor>
```

# Profiles

The Orbital Gateway includes functionality called Customer Profile Management, which allows Cardholder data to be stored with the Orbital Gateway. A merchant can process transactions by simply passing a token value that represents that cardholder.

Once a Profile is created, transactions can be processed, using either the on-line interface or the Orbital Virtual Terminal ["VT"], simply by referencing the Customer Profile and filling in any additional information not stored in the profile. This feature is only available to Merchants using the Chase Paymentech Orbital Interface.

## Not a recurring service

Currently, this feature is not a full recurring service. The Orbital Gateway will store all the relevant information for processing a transaction; it will not

automatically process it. Profile Management will still require the merchant to initiate a request to the Orbital Gateway and retrieve the result of that request.

## Benefits

There are potential benefits of using Profiles feature:

- It simplifies transaction processing. When making a transaction request, one simply references the Profile ID [a.k.a. the Customer Reference Number] and fills in any of the missing information.
- It eliminates risk. Since it eliminates the need to store sensitive information about your customer on your database, merchants can focus on their business, and Chase Paymentech can focus on securely processing your transactions.
- It can eliminate data entry errors when using the Virtual Terminal. By retrieving a pre-existing Profile and validating the data, it eliminates the risk of "keying" the wrong customer information such as Order # [which may equate to a Membership ID] or credit card #.

## Setup Information

For any Orbital Gateway Merchant ID to support Profiles, it must be configured on the Orbital System to do so. There are several different configuration aspects that must be setup.

- Enablement: First the Merchant ID must be configured to allow Profile functionality. Any Merchant ID that is not configured to use Customer Profiles and attempts to process a Profile Action will receive an error. A Profile Error Code of 9578 or "Merchant-Bin combination is not allowed to perform profile transactions" will be received.
- Customer Profile Hierarchy Support: Each Merchant ID must be configured to support Profiles at Chain ID (Company) level or Merchant ID level.
- Virtual Terminal Users: If your organization will utilize Profiles on the VT in addition to the XML interface, there are a few important considerations:

  **Profile User Management**

  - *Profile Administration:* For any VT User to administer Profiles [i.e. add, delete, update], that User must be provided the "right" to administer Profiles. Any existing User can be granted this additional User permission.
  - *Profile Usage:* For any VT User to use Profiles for processing a transaction, permission needs be granted to use profiles. Any existing User can be granted this additional User permission. The User will not be able to administer profiles, just use existing ones.
  - *Profile Disabled***:** If the VT User is not enabled for any Profile access level, they will not see any of the functionality. Profiles can be disabled for one user and enabled for another user.

  **General Access Rights:**

- *Card Masking:* The same card masking rules apply to Profile management or usage as they do to any card number viewing in the VT previously.
  - If a User's permission allows the viewing of the credit card number, then during usage or management, that User will be allowed to see any credit card number whether maintaining a profile or using it.
  - Conversely, if that User's permission level does not allow the number to be viewed, then it cannot be viewed whether they have the right to maintain a profile or use it. However, the card can be changed or updated regardless of masking.
- *Access Levels:* All existing access levels will not be impacted regardless of Profile user rights. For instance, if a User cannot submit credits, they will not be able to submit credits using Profiles.

## Business Rules

### How it works

Profile Management is a very simple product. The first step is to create a Profile. This can be done in two different fashions:

1. Adding a Profile as a distinct action
2. Or adding Profile as a part of an authorization request.

Once that Profile exists, it can be utilized to complete a sale or refund with any of the data elements stored in the profile. Additionally, any part of the Profile can be overridden during the subsequent transactions.

Finally, the Profile can be updated [or even deleted] at any point.

### Customer Reference Number Options

The Customer Reference Number [<CustomerRefNum> in the schema or 'Profile Number' in the Virtual Terminal] is the referential data element to a Profile.

Key Customer Reference Numbers facts:

- Must be unique [either by Merchant ID or Chain ID – see notes below].
- Can be from 1 to 22 bytes in length.
- Valid Characters are - abcdefghijklmnopqrstuvwxyzABCDEFGHIJKLMNOPQRSTUVWXYZ0123456789-,$@& and a space character.
- However, Alpha characters always stored in uppercase. Therefore, this

*Setting the Customer Reference Number*
The merchant can either set or request that the Orbital Gateway set the Customer Reference Number.

The field customerProfileFromOrderInd controls this behavior as follows:

A - Auto Generate the Customer Reference Number. In other words, the Orbital Gateway will assign the Customer Reference Number and return it in the response.

S - The Orbital Gateway will use the value passed in the <CustomerRefNum> Element as the Customer Reference Number.

O - This option only relates to when a Profile is added as a part of an authorization request. In this circumstance, the value passed in the <OrderID> element will be used as the Customer Reference Number. This would be used in circumstances wherein the order ID also represents the Customer ID in your system, such as a Policy Number for an insurance company.

D - This option only relates to when a Profile is added as a part of an authorization request. In this circumstance, the value passed in the <Comments> element will be used as the Customer Reference Number.

Note: Set field to "EMPTY" when using a customer profile. <CustomerProfileFromOrderInd="EMPTY">. This value is case sensitive.

*Using the Customer Reference Number to set other data elements*

The Orbital Gateway has options in configuring the Profile setup in terms of how the Profile ID can be leveraged to populate other data sets using the <CustomerProfileOrderOverideInd> value.

The options are:

NO - No mapping to order data.

OI - Pre-populate the following fields with the Customer Reference Number:

- <OrderID>
- <ECOrderNum> if it is an eCommerce Industry Type Transaction
- <MailOrderNum> if it is an Mail Order or Recurring Industry Type Transaction

OD - Pre-populate the <Comments> field [**Note**: this field is called Order Description in the Virtual Terminal] with the Customer Reference Number. The relevance of this feature is on the PNS platform [BIN 000002] the <Comments> field is what populates the Customer Definable Data. This data can then be made available on certain Resource Online Reports. Any questions about your reports should be directed to your Relationship Manager.

OA - Pre-populate the following fields with the Customer Reference Number:

- <OrderID>
- <ECOrderNum> if it is an eCommerce Industry Type Transaction
- <MailOrderNum> if it is an Mail Order or Recurring Industry
- <Comments>

### Customer Reference Number hierarchy setup and usage

As stated earlier, Profiles can be created at Merchant ID Level or at the Chain level.

If a MID is configured to use Profiles at a Chain ID level, any profiles setup by any Merchant ID are available to be used by any other Merchant ID's tied to that chain. However, if the MID is setup to manage Profiles at the merchant level, any Profile setup by that Merchant ID can only be used by that Merchant ID.

For example:

- Let's assume there is a single customer with two Merchant ID's on the Orbital Gateway, 11111 and 222222.
- These two merchant ID's are tied to the same Chain ID, 333333.
- Merchant ID 111111 sets up a new customer profile ABC.
- If both Merchant ID 111111 and Merchant ID 222222 are setup to manage Profiles at a chain level, then Merchant ID 222222 will be able to use profile ABC.
- If either one of them is not, the Merchant ID 222222 will not be able to use profile ABC.

Notes:

- All Profile Setups are performed at a Merchant ID. So this cross Chain ID sharing can only be facilitated via Orbital Setup.
- In addition, given that all setup and usage of Profile ID's is done using a specific Merchant ID, there is requirement that the Chain ID be known to take advantage if this feature. As long as all the Merchant ID's are properly linked to the same chain, it will simply work. If the Merchant ID's are not correctly mapped to the same Chain ID, Merchant ID's can be remapped to new Chain ID's easily. If this feature will be used, it is recommended that the correct chaining be validated prior to going live.
- Whatever level is defined as the Storage level, there can only be one version of a Customer Reference Number. Therefore if two different Merchant ID's have different customers who share the same Customer ID, it would be recommended that the Profile storage and usage be maintained at the Merchant ID level as opposed to the Chain level. If the second store tried to establish the same Customer Reference Number and the setup dictated a chain level storage, then a Duplicate Customer Reference Number error [<ProfileProcStatus> error code of 9582].

*Salem Hierarchy*

For Salem Orbital Gateway customers, the Orbital Gateway hierarchy closely emulates the Salem hierarchy.

- The Orbital Gateway MID will be the same as your Salem Division [or TD] number.
- And the Orbital Gateway Chain ID will be the same value as your Company Number [formerly known as the MA].

If the Salem Division numbers are all linked to a specific Company number that is in fact how it will be setup on the Orbital Gateway.

*PNS Hierarchy*
For PNS Orbital Gateway customers, the Orbital Gateway hierarchy is tied to the PNS Authorization Host hierarchy. As such:

- The Orbital Gateway MID will be the same as your PNS Authorization Merchant ID [MID] – Terminal ID [TID].
- However, there is no PNS Chain value. So the Orbital Gateway assigns the next available chain value when setting up accounts for the first time.

If an organization has multiple Merchant IDs, there is no guarantee that all of those Orbital Gateway Merchant ID's will be linked under a single Chain ID. However, Merchant IDs can be moved under one chain to take advantage of this feature.

### Profile Transaction Types

There is a number of transaction types associated with Profiles. Some of these are extensions of existing transaction types and some are new to profiles. This section will detail from an XML perspective how to support all those Profile transaction types and some of the specific rules associated with each of them. Again, all of the functionality identified within this document is possible through the Virtual Terminal as well.

*Managing Profiles*
There are a set of transactions specifically set up for managing the Profile; in other words, adding, updating, deleting and retrieving the information.

Adding Profiles
First and foremost, a profile needs to be added to the Orbital Gateway. There are two different transaction actions that can be performed to add a profile.

*Adding Profiles as a Stand Alone Transaction*
The simplest mechanism is to add a Profile is to simply make a Profile Add Request Type. This document includes both the definition of the XML elements necessary to complete this transaction and an example template of an Add Profile Request.

There are new response data elements that need to be interpreted to determine the success of this Add request.

*Adding Profiles during an Auth*
Given that in many circumstances, the fist time a customer is setup an authorization needs to be performed, the Orbital Gateway has extended the traditional Authorization transaction to enable adding a Profile in the same request.

- Any data included in the Authorization that can be saved as a part of the profile will be.
- The minimum data to create a profile must be included or no profile will be created.

- The result of the authorization is separate from the result of the profile add step. As such, an authorization can be successful and the Profile Add component can be an error on the same transaction and vice-versa. These results are mutually exclusive and should be interpreted from a response management process as such.
- Add profiles can only be included with Auth Only, Auth-Capture and Prior Auth [i.e. Force] transactions. It cannot be completed as a part of a Refund, Void, or Mark for Capture.

*Information saved in a profile*

Whether a Profile is added via a Profile Add transaction or added via an authorization or updated later via a Profile Update transaction, the following list of items defines what data elements can be saved as a part of a Profile.

- Customer Reference Number – Required and un-editable (Also referred to as Profile ID)
- Customer Name
- Customer Email – NOTE: Only Available as a Profile Add or Update. This XML element does not exist yet in the Authorization message set and cannot be sent as a part of an Auth Request profile Add as a result.
- Address Information:
    - Address 1
    - Address 2
    - City
    - State
    - ZIP
    - Phone
- Amount
- Description: This can be set in two ways. Either by sending a specific description message or by setting the <CustomerProfileOrderOverideInd> to populate the <Comments> tag, which is the Order Description.
- All Order # fields: This can be accomplished by setting the <CustomerProfileOrderOverideInd>
- Payment Information:
    - Card Type:
        - Credit Card
            - Card #
            - Expiration Date
        - ECP (Salem Host Only – BIN 000001)
            - DDA
            - R/T
            - Account Type
            - Payment Delivery Method
        - Switch Solo (Salem Host Only – BIN 000001)
            - Card #
            - Expiration Date
            - Start Date
            - Issue Number

### *Information NOT saved in a profile*

There are a number of items that will not be added to Profile regardless of how it is done. This includes but is not limited to:

- Purchasing Card Data
- Card Verification Number [CVV2, CVC2, and CID]. This is because this information cannot be stored by card association regulation. It must be requested from a cardholder on a transaction-by-transaction basis.
- VbV and MasterCard SecureCode Data

### Updating Profiles

Once a Profile has been added, any information about a Profile can be updated, except the Profile Keys [which again includes the Customer Reference Number, Merchant ID, and BIN].

This can be accomplished by sending a Profile Update transaction.

Some important keys to performing an Update:

- All Profile Update requests must include the correct Profile keys or an error message will be returned. (A list of the error messages can found in the appendix)
- An update requires the Tags to be sent for both:
    - The Data that should be updated.
    - And for any fields that should be cleared.
- To clear any legacy data, the XML Tag is submitted with nothing but a Tilde ['~']. See below example for performing this step:

<CCExpireDate>~</CCExpireDate>

- For example, if the Customer Profile included an amount and update was sent with the Amount Tag present, but filled with a Tilde character, the amount stored in the profile would be changed to null in the database.
- If an XML tag is sent with a Null value [such as <CCExpireDate></CCExpireDate>], it will be ignored as a part of the update process [i.e. no update would occur on the CCExpireDate value from the example].
- When changing Card Types, such as from an ECP to a Credit Card what is required is:
    - Send the XML tag representing the new card type
    - Submit the appropriate data for that card type.
    - Null the old card type data elements using the Tilde process described above. For example, changing from an ECP transaction type to a Credit Card type, the Profile Update message should:
        - Have the Card Type defined as Credit.
        - The Update message should include the Credit Card # and Exp Date
        - And it should send a Tilde for the four ECP data elements [DDA, R/T, Account Type, and Payment Delivery Method]

### Retrieving a Profile

At any given time, there may be a need to retrieve the data on an existing Profile. There is a very simple Retrieve Profile transaction type available to perform this action.

### Deleting a Profile

Any Profile can be deleted at any time with a Delete Profile transaction.

At this stage, even after a Profile has been deleted a Customer Profile Reference Number still may not be used again.

### Using Profiles

One of the key transaction types is using a Profile to process a transaction. This is accomplished by:

- Inserting the Customer Reference Number into one of the existing message types.
- All data that can be pre-populated by the Profile will be.
- Any relevant data, such as CVD for eCommerce transactions, should be included in the request.
- The transaction request should be completed per the normal spec in terms of which tags are mandatory. If the data exists in the Profile and the Tag is mandatory, simply null fill the tag.
- This means that the correct XML Message type should be used based on the card type of the profile. For example, if the card type of a Profile is a credit card, then the base credit card message structure should be used to use the profile. The credit card data, again, would simply be null filled.

### Overriding Profile Data

Almost any data set in the Profile can be overridden during a transaction that is using the Profile. For instance, if a Profile included a fixed amount, but a particular transaction was for a different amount, it could be changed for that transaction by including a specific amount in the use profile request.

The one exception to the override rule is the payment type, such as Credit Card versus ECP, cannot be overridden. If the Payment is different, then the Profile should either be Updated [if that change is permanent] or not used [if it is temporary].

By the same token, if the payment type is the same, but the data is different, it can be overridden on a single transaction, if desired.

Finally, overriding Profile data does not update the profile. If the change is permanent, an Update Profile request should be sent in.

#### *Expiration Date*

One scenario to take into consideration when overriding data has to do with the usage of expiration dates. As defined in the spec, for a Salem customer, a null Expiration date is one mechanism to submit transactions for authorization when the expiration date is unknown. By the same token, Expiration Date is a required tag for credit card transactions and must be present when Using Profile. And it must be null filled to not override the expiration date that might be set in the profile.

As such, if an Expiration date is saved in a Profile, and the desire is to override that, but submit nothing because the new expiration date is unknown, the transaction should use one of the other two supported mechanisms for supporting unknown expirations dates:

- Send four spaces - <Exp>  </Exp>
- Zero fill the XML Element - <Exp>0000</Exp>

Transaction Types

Profiles may be used on Authorization, Authorization-Capture, Prior Authorizations, and Refund transactions. It is not functional (or necessary) for Voids, Mark for Capture, or End of Day transactions.

*Mark for Capture Transactions*

While the correct Credit Card or Switch / Solo Mark for Capture ['MFC'] transaction never really required the card number from a business perspective, the Orbital Gateway DTD mandated the presence of the XML element. So many customers are either sending the actual credit card number, a dummy number or null filling this tag. Effective with the release of Profile Management, the Orbital Gateway has corrected this design and no longer mandates this tag to be a part of the MFC request. This was done to ensure that a customer using a Profile could complete a MFC without having to include this tag.

Industry Types

All the Industry Types are supported by the Orbital Gateway [eCommerce, Mail Order, and Recurring] are all supported within Profiles. The transaction in question should set the correct tags as identified in the XML Message Specification.

Currencies

All currencies supported by the Orbital Gateway are supported as a part of Profiles. Simply set the correct currency code and exponent on the transaction being processed.

### Retry Logic Usage

Retry Logic, the function that allows transactions to be processed without risk of duplicating them will not be supported for Profile Management transactions. In other words, Adds, Deletes, Retrieves and Updates.

However, if when performing a Profile Management transaction, an unknown result occurs, simply replay that transaction.

- If the prior transaction was a success, the second attempt will simply result in duplicate response, which not cause any harm.
- If the original request was not successful, the second attempt will create the result desired.

While Retry is not supported for Profile Management transactions, there is no harm in placing the Trace-ID values associated with Retry Logic in the MIME-Header of these request items. In these circumstances, the trace value will simply be ignored.

**Note:** Using Profile during an Authorization, Retry Logic is fully supported as defined the message specification.

# *Processing Interface Description*

## Introduction

The Processing Interface for the Chase Paymentech Gateway is logically defined as follows:

The Chase Paymentech API uses XML (Extensible Markup Language) to make e-commerce payment requests using HTTPS (Hypertext Transfer Protocol Secure). It allows you to submit all transaction types supported by the Orbital Gateway, such as authorization, authorization and capture, prior authorization, capture, refund, void, and an end of day (batch).

## Communication Protocol

The Chase Paymentech gateway supports one method of communication – HTTPS. This method provides a 'single-threaded' (or synchronous) model in which a merchant makes an HTTPS request to the gateway and then blocks until the gateway sends back HTTPS response. While HTTPS is single threaded, a single interface can make multiple HTTPS requests at once.

### Posting to a URL

The Orbital Gateway will only provide responses to HTTP POST requests. The POST method is used to request that the origin server accept the entity enclosed in the request as a new subordinate of the resource identified by the Request-URI in the Request-Line. Orbital Gateway does not support GET requests.

The addresses for the certification and production Orbital Gateway are:

**Orbital Gateway certification system:**

Primary: 'orbitalvar1.paymentech.net/authorize' on port 443

Secondary: 'orbitalvar2.paymentech.net/authorize' on port 443

**Orbital Gateway production system**:

Primary: 'orbital1.paymentech.net/authorize' on port 443

Secondary: 'orbital2.paymentech.net/authorize' on port 443

Note:

- Chase Paymentech exposes redundant hostname/port network endpoints to ensure high availability for the Orbital Gateway. Developers should code to

a fail over URL. When connectivity fails, fail-over to the secondary hostname/port should be automatic. Communication with the primary hostname/port should be attempted periodically. The fail over should be automatic and completely transparent to end-user.

- Caching IP Addresses of Orbital Gateway servers is strongly discouraged. For load balancing and redundancy reasons, the Orbital Gateway processing is divided amongst multiple data centers. Therefore, the DNS service should be used to determine the destination IP address for each transaction.
- While the certification system is available for testing at all hours, it is only monitored for availability during business hours [8:00am EST – 5:30pm EST Monday – Friday].
- In addition, the hardware in place is designed primary for certification testing, not load testing. If there is a need to ensure uptime outside of normal business hours, please advise your certification analyst of the testing requirements.

# Security

Given the inherent risks associated with processing transactions over the Internet, the Orbital Gateway requires both encrypted traffic to prevent interception of the Payload and authentication of the source request generation. The next two sections define how the system manages that security.

## Secure Sockets Layer Implementation Required

The XML gateway URL must be accessed using the **`https`** protocol so that private information is transferred securely. This requires the client to use a SSL implementation. There are SSL implementations available for most programming languages.

**It is the client's responsibility to gain the necessary expertise and technology to properly open a secure channel to the gateway [unless the client uses of one Chase Paymentech's available SDK's].**

Interfacing to the Orbital Gateway using SSL does not require the client to have a certificate. The Orbital Gateway uses a non-authenticated SSL session, meaning the client is not authenticated using a digital certificate as a component of the SSL negotiation. See below section for how Chase Paymentech authenticates client traffic.

Non-SSL postings should never be made across a network that is external or not totally controlled and secure. If a clear text request is made to one of the Orbital Gateway URL's, it will return an error condition - an HTTP 403 error with the accompanying XML payload containing a ProcStatus 20403 error.

## Authentication

The Orbital Gateway uses source IP authentication to authenticate the request generation. What this means from a client implementation is as follows:

- When processing transactions against both the Test and Production Orbital Gateway, the client servers Source IP or Source IP's must be registered in the Orbital Gateway. Any activity presented on an IP address that is not registered in the Orbital Gateway will result in an HTTP 412 error with the accompanying XML payload containing a ProcStatus 20412 error – see documentation for definition of these error fields].
- In addition, these IP addresses must be affiliated the Merchant ID's for which the Client should be submitting transactions. Specifically:
    - This does allow Third Party Hosting service organizations presenting on behalf of other merchants to submit transactions. However, each time a new customer is added, the merchant or Third-Party hosting organization needs to ensure that the new Merchant ID's or Chain ID's are affiliated with the hosting companies IP's.
    - If the merchant expects to have more than one merchant account with the Orbital Gateway, it should have its IP addresses affiliated at the Chain level hierarchy within the Orbital Gateway. Each time a new merchant ID is added, as long as it is placed within the same Chain, it will simply work. Otherwise, the additional MID's will need to be affiliated with the merchant IP's. For example, we generally affiliate all Salem accounts [BIN 000001] with their Company Number [formerly called MA #] number so all MIDs or Divisions under that Company will automatically be affiliated.
- IP Failure and MID Failure
    - If an IP is registered, but the client presents an MID that has NOT been associated with the originating IP, the Orbital Gateway will return the following error: HTTP 200 with a ProcStatus of 9717.

# XML Schema

The Orbital Gateway accepts and returns XML documents using XML Schema Definition (XSD) that are defined by Chase Paymentech. The latest XSD published for interfacing with the Orbital Gateway is 'PTI41'. There are separate request and response XSD's [Request_PTI41.xsd & Response_PTI41.xsd].

Note: Versions 'PTI40' and above are not backward compatible with XML documents created in DTD version created in PTI34 and earlier. All prior DTD versions are still supported but require the Mime-Header to identify the version as PTI34 or lower. Any new functionality exposed after PTI34 requires coding to the new XSD specification.

This allows Chase Paymentech requests and responses to be easily interpreted and manipulated using the w3c (World Wide Web Consortium) DOM (Document Object Model) or SAX (Simple API for XML) APIs.

# MIME Header

MIME (Multipurpose Internet Mail Extensions) is a mechanism for specifying and describing the format of Internet message bodies. The Orbital Gateway supports both the HTTP/1.0 and HTTP/1.1 MIME Header specifications for describing the message payload along with other information that allows it to process the incoming transaction request; as well as the outgoing reply.

## MIME Header Field Definition

### Request MIME Header Definition:

The following table lists each of the required fields within the MIME-Header for Orbital Gateway requests.

| Field | Definition | M/C |
|---|---|---|
| POST /AUTHORIZE HTTP/1.0 | Fixed Header | M |
| MIME-Version: | Should always be "1.0" or "1.1". | M |
| Content-type: | Defines the XML version number. See more definition below. | M |
| Content-length: | This value defines the length of the Request XML Document | M |
| Content-transfer-encoding: | Defines the encoding of the associated XML document. Recommended encoding is "text". | M |
| Request-number: | Should always be "1" | M |
| Document-type: | Defines whether this is a Request on Response. This value should always be "Request". | M |
| Merchant-id: | Merchant ID. See definition below. | C |
| Trace-number: | Retry Trace-number. See definition below. | C |
| Interface-Version: | Optional MIME-Header element that can be used by Chase Paymentech in production support. See more definition below. | O |

*POST /AUTHORIZE HTTP/1.0:*
This value is static and always should be presented as indicated. The details of this are as described in the following sections.

- HTTP Post: The Orbital Gateway will only provide responses to HTTP POST requests
- URI: The Request URI for the Orbital Gateway will always be Authorize.
- HTTP 1.0 or 1.1: Provide the HTTP Version #.

*Content-Type:*
This MIME-Header field is used by the Client to identify which DTD version is being used for the XML Payload.

- The format data for this field is "application/(XSD Version)".
- The latest Version and recommended value is PTI41.
- As such, the recommended value for this field is "application/PTI41"

  NOTE: Versions 'PTI40' and above are not backward compatible with XML documents created in DTD version created in PTI34 and earlier. All prior DTD versions are still supported but require the Mime-Header to identify the version as PTI34 or lower. Any new functionality exposed after PTI34 requires coding to the new XSD specification.

*Merchant-id:*
This MIME-Header field is used in combination with the Trace-number as the key transaction identifiers as it relates to Retry Logic. This field must be populated with the same Merchant ID as submitted in the XML payload, i.e. [<MerchantID>123456789012</MerchantID>]. If the Merchant ID is not the same, the Orbital Gateway will return an error in the form of a Quick Response with a ProcStatus of 9713, "Invalid mime header - Merchant ID in mime does not match XML message" [see section 4 for complete description of Retry Logic specific error codes].

*Trace-number:*
This MIME-Header field is used in combination with the Merchant-id as the key transaction identifiers as it relates to Retry Logic. The Trace-number Field rules are:

- Data Type = Numeric
- Minimum Length = 1
- Maximum Length = 16
- As such, Valid Values are from 1 – 9999999999999999
- Submitting an invalid value will result in an XML Quick Response with a ProcStatus Code of 9714.

*Interface-Version:*
This MIME-Header field is used by the Client to identify information about their implementation to assist Chase Paymentech in providing production support. This information will be logged distinctly for research purposes. An example of the usage of this filed would be any Third Party Software Provider should log

their Software name and version number in this field, such that Chase Paymentech knows how the interface is being managed. A Merchant could place information about their development version and implementation language. Please work with your Certification Analyst to best identify what values should be used in this field.

### Response MIME Header Definition:

This may not be an all-inclusive list of MIME-Header response fields. Do not code your system to support only these.

| Field | Definition | M/C |
|---|---|---|
| HTTP | See below section for more definition | M |
| Date: | Returns the Server Date and Tamp stamp, for example: Fri, 27 Oct 2000 20:29:58 GMT | M |
| MIME-Version: | Will always be "1.1". | M |
| Content-type: | Defines the XML version number. This will be an echo of what is submitted in the request | M |
| Content-length: | This value defines the length of the Response XML Document | M |
| Content-transfer-encoding: | Defines the encoding of the associated XML document. | M |
| Request-number: | Should always be "1" | M |
| Document-type: | Defines whether this is a Request or Response. This value will always be "Response". | M |
| Resend-Count: | Identifies the number of times a response is returned. See below section for more definition. | C |
| Last-Retry-Attempt: | Identifies the last previous retry response sent. See below section for more definition. The format of the data is:<br>YYYYMMDDhh(24)mmss | C |

### HTTP Responses

When successfully interacting with the Orbital Gateway, the HTTP value returned will always be a 200 response, such as "HTTP/1.0 200 OK". All other responses indicate some sort of connection problem. Please refer to RFC # 2616 http://www.w3.org/Protocols/rfc2616/rfc2616.txt for more definition of the variance HTTP responses that could be returned and their meaning.

A HTTP 200 response in and of itself does not constitute a good response. It simply means that the connection has successfully been established with the

Orbital Gateway. Please read the response handling sections for full response interpretation definition.

#### Resend-Count:

The purpose of this MIME-Header Response Field is to expose how frequently, the Gateway has returned a particular response to your system. The first time a transaction is submitted to our system, the Resend-Count will be "0". The first time a transaction is Retried, the Resend-Count will be "1".

If a value of greater than "1" is returned, the Gateway is returning the same result many times for the same transaction. It can be an indicator that a customer is un-intentionally replaying the same transaction or having trouble reading the result.

#### Last-Retry-Attempt:

This MIME-Header response field is returned if the Resend Count is greater than or equal to two [2]. In other words, it will be returned as soon as a second Retry Response is sent and all others thereafter. It identifies the date and time, the last time a Retry Response for the associated Trace-Number and Merchant ID was returned. It provided as an additional mechanism to ensure that the Retry function is behaving as expected.

# Retry Logic

## Business Rules

Retry Logic is a function available from the Orbital Gateway for client interfaces to reprocess transactions when there is an unknown result on an XML transaction request. It is available to any merchant interfacing to the Orbital Gateway using XML by simply adding two new values to the MIME-Header, the Merchant ID and a transaction Trace-number. The Orbital Gateway uses this combination of values to determine the uniqueness of a transaction in determining how to process the transaction.

The basic process flow of Retry Logic is as follows:

- A Request is submitted with a Trace-number and Merchant ID in the MIME-Header.
- The Gateway validates the Trace-Number and Merchant ID to determine if it has processed a transaction using that value pair within a 48-hour window.
- If the transaction is a decline or error on the initial response, the next request will be treated as a new request.
- If it has not processed the pair, the Gateway will treat that transaction as a new request and process it accordingly.
- If it has processed the pair and the Request has either already been processed [the initial response is an approval] or is in process, the Orbital Gateway will

immediately echo back the exact response XML Document from the initial request.
- If the initial Request is still in process, the Orbital Gateway will block and wait until that original response is completed. As soon as that is done, it will then echo back the same response as the original request.

The following sections outline the detail business rules and implementation considerations associated with Retry Logic.

## Retry Timing

The Orbital Gateway only retains an original Retry Trace-number, Merchant ID pair for 48 hours after submission. Any transactions that reuse these values more than 48 hours after the original transaction was submitted will be treated as a new request.

Therefore, if there is an unknown result for a transaction, then that transaction must be reattempted within 48 hours or the result will need to be determined through the Virtual Terminal Interface prior to regenerating the transaction.

## XML Request Validation on Duplicate TRACE-Numbers

The following is a description of the message validation of the XML document when the Retry Attempt is made that matches a prior Trace-number, Merchant ID combination.

- The following conditions will result in an Error, even if the Trace-number, Merchant ID Combination is a match:
  - There is no XML Document Present.
  - That XML Document does not pass Schema validation based on the version number passed in the MIME-Header.
  - The Merchant ID in the XML Document does not match the Merchant ID in the MIME-Header.
  - The Request Type [i.e. Auth versus an Auth Capture versus Refund, etc] must be the same. If the request type changes between transactions, a Quick response will be returned with a ProcStatus of 9715.
- No other validation is associated with the XML Document of this request, meaning:
  - If that XML Document is not the same as the XML Document associated with the original request using that Trace-number & Merchant ID pair, Retry Logic will still echo back the original response XML Document. It does not matter if the credit card number is different or if the request type is different altogether.
    - As such, it is very important when implementing Retry Logic that the Trace-number process is implemented correctly. Otherwise, the same result could be returned for different requests multiple times.

**Important Note**

**If the Trace-number & Merchant ID pair do not match a prior transaction in the previous 48 hour window, the Orbital Gateway will treat that new XML Document as a new request and deal with it accordingly even if it were a "duplicate" transaction.**

## Transaction Types Supported

The Retry Logic for initial transactions and retry attempts can be used for all transactions types.

## Retry Error Responses

When an error occurs resulting from the clients implementation of Retry Logic:

- That XML request will not be processed.
- An error will be returned in the form of a Quick Response in an XML Document, just as various other Orbital Gateway errors are returned.

## Concurrency

There is no limit to the number of Retry attempts on a transaction, as long as they all occur within the 48-hour window.

However, no more than two concurrent transactions with the same Trace-number / Merchant ID value pair can be in process with the Orbital Gateway at any given time. If more than two transactions are sent while the Orbital Gateway is in the midst of processing the first two, it will immediately respond in the form of a Quick Response with a ProcStatus of 9711, "Too many transactions to process.

If this occurs, it might be an indicator of a Client Problem. There would never be a reason to have more than two concurrent requests in queue with the same Trace-number on a particular MID. As such, receiving this response code could indicate that the Trace-number is not always being generated uniquely when it should be or your system is not waiting for responses long enough.

## Merchant ID

Retry Logic requires that the Merchant ID be submitted in the MIME-Header in addition to the XML Document. The Merchant ID's submitted in the MIME-Header and the XML Document must be the same or the Orbital Gateway will return an error in the form of a Quick Response with a ProcStatus of 9713, "Invalid mime header - Merchant ID in mime does not match XML message".

## Retry Attempt Time Out

As indicated above, when a Retry Attempt is made while the Original Request is still in process, the Orbital Gateway will block and wait for that original response

to be created with the intent to echo that completed response in the Retry Response. However, the Orbital Gateway must return a result to the Client on all requests in no more than 90 seconds, including a Retry Attempt. Therefore, there is a time limit on how long the Retry attempt will block and wait. If the Original Request response is not complete prior to this window, a Quick Response ProcStatus of 9710, "Message expired during retry", will be returned.

If this occurs, the correct action is to make a second Retry Attempt of the transaction with the Original requests Trace-number, Merchant ID Pair.

# *Transaction Management Messages*

## New Order Request Elements

Auth, Auth Capture, Prior Auth/Capture and Refunds

| XML Name | XML Parent Element | Description | Req. M/C/O | Max. Char. | Field Type A/N |
|---|---|---|---|---|---|
| **Request** | N/A | **Required XML Parent Tag** | M | N/A | N/A |
| **NewOrder** | Request | **XML tag that defines the transaction as a New Order request** | M | N/A | N/A |
| **IndustryType** | NewOrder | **Defines the Industry type of the transaction:**<br>MO – Mail Order transaction<br>RC – Recurring Payment<br>EC– eCommerce transaction<br>IV – IVR [PINLess Debit Only] | M | 2 | A |
| **MessageType** | NewOrder | **Defines the transaction New Order Transaction Type:**<br>A – Authorization request<br>AC – Authorization and Mark for Capture<br>FC – Force-Capture request<br>R – Refund request | M | 2 | A |
| **BIN** | NewOrder | **Transaction Routing Definition:**<br>Assigned by Chase Paymentech<br>000001 – Salem<br>000002 – PNS | M | 6 | N |

| XML Name | XML Parent Element | Description | Req. M/C/O | Max. Char. | Field Type A/N |
|---|---|---|---|---|---|
| **MerchantID** | NewOrder | **Gateway merchant account number assigned by Chase Paymentech:**<br>This account number will match that of your host platform<br>BIN 000001 – 6 digit Salem Division Number<br>BIN 000002 – 12 digit PNS Merchant ID | M | 12 | N |
| **TerminalID** | NewOrder | **Merchant Terminal ID assigned by Chase Paymentech:**<br>All Salem Terminal IDs at present must be '001'. PNS Terminal ID's can be from '001' – '999'. Most are '001'. | M | 3 | N |
| **CardBrand** | NewOrder | **Defines the Card Type / Brand for the Transaction:**<br>See attached table.<br>Required for:<br>SW – Switch / Solo<br>ED – European Direct Debit<br>EC – Electronic Check<br>BL – Bill Me Later<br>DP – PINLess Debit [Generic Value Used in Requests] | C | 2 | A |
| **AccountNum** | NewOrder | **Card Number identifying the customer.**<br>Should be null for electronic check processing and Profile Transactions<br>For Bill Me Later transactions, the account number field should be populated with either the customer's Bill Me Later account number or a Bill Me Later Bank Identification Number (BIN) followed by ten zeros (dummy account number) – such as 5049900000000000. The consumer's 16-byte Bill Me Later account number will be returned on all approved transactions. | M | 19 | N |

| XML Name | XML Parent Element | Description | Req. M/C/O | Max. Char. | Field Type A/N |
|---|---|---|---|---|---|
| **Exp** | NewOrder | **Card Expiration Date:**<br>- Formats: MMYY<br>- Mandatory for all card types except ECP, European Direct Debit, Bill Me Later and PINLess Debit; except as defined below:<br>Salem [BIN 000001] allows a blank to be submitted when no known EXP date exists. Please discuss this feature with your certification analyst before implementing. There are three valid mechanisms for submitting a 'Blank' expiration date using Orbital to the Salem Host. They are:<br>- Null fill this XML element - <Exp/><br>- Send four spaces - <Exp> </Exp><br>- Zero fill the XML Element - <Exp>0000</Exp> | C | 4 | N |
| **CurrencyCode** | NewOrder | **Defines the transaction currency:**<br>The gateway using the standard ISO defines currency codes. Keys:<br>- Bin 000002 only supports the US Dollar ['840'] and Canadian Dollar ['124'].<br>- See Appendix for complete list of supported currencies. | M | 3 | N |
| **CurrencyExponent** | NewOrder | **Defines the transactions currency exponent:**<br>See Appendix for values. | M | 6 | N |
| **CardSecValInd** | NewOrder | **Supported by Visa and Discover Only:**<br>1 - Value is Present<br>2 - Value on card but illegible<br>9 - Cardholder states data not available<br>NOTE: If the transaction is not a Visa, MasterCard or Discover transaction, null fill this attribute or do not submit the attribute at all. | C | 1 | N |

| XML Name | XML Parent Element | Description | Req. M/C/O | Max. Char. | Field Type A/N |
|---|---|---|---|---|---|
| **CardSecVal** | NewOrder | **Card Verification Number**<br>- Visa CVV2 – 3 bytes<br>- MasterCard CVC2 – 3 bytes<br>- American Express CID – 4 bytes<br>- Discover CID – 3 bytes<br>NOTE: It is against regulations to store this value. | O | 4 | N |
| **DebitCardIssueNum** | NewOrder | **Switch – Solo incremental counter for lost or replacement cards.**<br>- Tag Mandatory for Switch – Solo.<br>- An incremental counter of either 1 or 2 characters defined by the issuing bank. If a card is lost, the bank issues a replacement card with the issue number being increased by one. If the card displays '01', submit '01', NOT '1'. If the card displays '1', submit '1'.<br>NOTES:<br>- The DebitCardStartDate field should be submitted only when the card does not have an Issue Number. If the card displays ONLY a Start Date and no Issue Number, the DebitCardStartDate field should contain a value and the DebitCardIssueNum field must be left blank [null filled]. If the card displays both a Start Date and an Issue Number, the DebitCardStartDate field should be left blank [null filled] and the DebitCardIssueNum field must be populated. | C | 2 | N |

| XML Name | XML Parent Element | Description | Req. M/C/O | Max. Char. | Field Type A/N |
|---|---|---|---|---|---|
| **DebitCardStartDate** | NewOrder | **Switch – Solo card start date:**<br>- Format 'MMYY'<br>- Tag Mandatory for Switch – Solo.<br>NOTE: The card start date should be submitted only when the card does not have an Issue number. If the card displays ONLY a Start Date and no Issue Number, the DebitCardStartDate field must be blank [null filled]. If the card displays both a Start Date and an Issue Number, this field should be left blank and the DebitCardIssueNum field must be populated. | C | 4 | N |
| **BCRtNum** | NewOrder | **Bank routing and transit number for the customer.**<br>Conditionally required for Electronic Check processing.<br>NOTE:<br>- All US Bank Routing Numbers are 9 digits<br>- All Canadian Bank Routing Numbers are 8 Digits | C | 9 | N |
| **CheckDDA** | NewOrder | **Customer DDA account number**<br>Conditionally required for Electronic Check processing. | C | 17 | A |
| **BankAccountType** | NewOrder | **Defines the deposit account type:**<br>Conditionally required for Electronic Check processing.<br>C – Consumer Checking [US or Canadian]<br>S – Consumer Savings [US Only]<br>X – Commercial Checking [US Only] | C | 1 | A |
| **ECPAuthMethod** | NewOrder | **Defines the ECP Authorization method:**<br>Code used to identify the method used by consumers to authorize debits to their accounts. If no value submitted, we will default this value.<br>Valid Values:<br>W – Written<br>I – Internet (Web) – DEFAULT<br>T – Telephone | O | 1 | A |

| XML Name | XML Parent Element | Description | Req. M/C/O | Max. Char. | Field Type A/N |
|---|---|---|---|---|---|
| **BankPmtDelv** | NewOrder | **Defines the ECP payment delivery method:**<br>This field indicates the preferred manner to deposit the transaction.<br>Conditionally required for Electronic Check processing.<br>B – Best Possible Method [US Only]<br>Chase Paymentech utilizes the method that best fits the situation. If the RDFI is not an ACH participant, a facsimile draft will be created. This should be the default value for this field.<br>A – ACH [US or Canadian]<br>Deposit the transaction by ACH only. If the RDFI is not an ACH participant, the transaction is rejected. | C | 1 | A |
| **AVSzip** | NewOrder | **Cardholder Billing Address Zip Code:**<br>- All AVS Requests must minimally include the 5-digit Zip Code.<br>- If sending Zip Code + 4, please separate with a '-'<br>Required for Bill Me Later sale transactions | C | 10 | A |
| **AVSaddress1** | NewOrder | **Cardholder Billing Address line 1**<br>Should not include any of the following characters types:<br>% \| ^ \ /<br>Required for Bill Me Later sale transactions | C | 30 | A |
| **AVSaddress2** | NewOrder | **Cardholder Billing Address Line 2**<br>Should not include any of the following characters types:<br>% \| ^ \ / | O | 30 | A |
| **AVScity** | NewOrder | **Cardholder Billing City**<br>Should not include any of the following characters types:<br>% \| ^ \ /<br>Required for Bill Me Later sale transactions | C | 20 | A |

| XML Name | XML Parent Element | Description | Req. M/C/O | Max. Char. | Field Type A/N |
|---|---|---|---|---|---|
| **AVSstate** | NewOrder | **Cardholder Billing State**<br>Should not include any of the following characters types:<br>% \| ^ \ /<br>Required for Bill Me Later sale transactions | C | 2 | A |
| **AVSphoneNum** | NewOrder | **Cardholder Billing Phone Number**<br>AAAEEENNNNXXXX, where<br>AAA = Area Code<br>EEE = Exchange<br>NNNN = Number<br>XXXX = Extension<br>Required for Bill Me Later sale transactions | C | 14 | A |
| **AVSname** | NewOrder | **Cardholder Billing Name**<br>Required for Bill Me Later sale transactions and all Electronic Check Transactions. | C | 30 | A |
| **AVScountryCode** | NewOrder | **Cardholder Billing Address Country Code:**<br>Required if processing a U.K. based Address. Valid values:<br>US – United States<br>CA – Canada<br>GB – Great Britain<br>UK – United Kingdom<br>" " – Blank for all other countries<br>Required for Bill Me Later sale transactions | C | 2 | A |
| **AVSDestzip** | NewOrder | **Bill Me Later Cardholder Destination Address Zip Code:**<br>- All AVS Requests must minimally include the 5-digit Zip Code.<br>- If sending Zip Code + 4, please separate with a '-'<br>Required for Bill Me Later sale transactions | C | 10 | A |

| XML Name | XML Parent Element | Description | Req. M/C/O | Max. Char. | Field Type A/N |
|---|---|---|---|---|---|
| **AVSDestaddress1** | NewOrder | **Bill Me Later Cardholder Destination Address line 1**<br>Should not include any of the following characters types:<br>% \| ^ \ /<br>Required for Bill Me Later sale transactions | C | 30 | A |
| **AVSDestaddress2** | NewOrder | **Bill Me Later Cardholder Destination Address Line 2**<br>Should not include any of the following characters types:<br>% \| ^ \ /<br>Optional for Bill Me Later Transactions | O | 30 | A |
| **AVSDestcity** | NewOrder | **Bill Me Later Cardholder Destination Billing City**<br>Should not include any of the following characters types:<br>% \| ^ \ /<br>Required for Bill Me Later sale transactions | C | 20 | A |
| **AVSDeststate** | NewOrder | **Bill Me Later Cardholder Destination Billing State**<br>Should not include any of the following characters types:<br>% \| ^ \ /<br>Required for Bill Me Later sale transactions | C | 2 | A |
| **AVSDestphoneNum** | NewOrder | **Bill Me Later Cardholder Destination Phone Number**<br>AAAEEENNNNXXXX, where<br>AAA = Area Code<br>EEE = Exchange<br>NNNN = Number<br>XXXX = Extension<br>Optional for Bill Me Later sale transactions | O | 14 | A |
| **AVSDestname** | NewOrder | **Bill Me Later Cardholder Destination Billing Name**<br>Required for Bill Me Later sale transactions | C | 30 | A |

| XML Name | XML Parent Element | Description | Req. M/C/O | Max. Char. | Field Type A/N |
|---|---|---|---|---|---|
| **AVSDestcountryCode** | NewOrder | **Bill Me Later Cardholder Destination Address Country Code:**<br>Required if processing a U.K. based Address. Valid values:<br>US – United States<br>CA – Canada<br>GB – Great Britain<br>UK – United Kingdom<br>" " – Blank for all other countries<br>Required for Bill Me Later sale transactions | C | 2 | A |
| **CustomerProfileFromOrderInd** | NewOrder | **Customer Profile Number generation Options:**<br>A – Auto Generate the CustomerRefNum<br>S – Use CustomerRefNum Element<br>When performing an action other than a Customer Profile Action Type = Create, insert "empty" as it will be ignored. But this Attribute is mandatory and cannot be null filled for Customer Profiles. | M | 5 | A |

| XML Name | XML Parent Element | Description | Req. M/C/O | Max. Char. | Field Type A/N |
|---|---|---|---|---|---|
| **CustomerRefNum** | NewOrder | **Sets the Customer Reference Number that will be used to utilize a Customer Profile on all future Orders.**<br><br>Mandatory if Customer Profile Action Type = Read, Update, or Delete<br><br>Or<br><br>Create and the Customer Profile Number generation option = S [Use CustomerRefNum Element]<br><br>Keys:<br><br>- If CustomerProfileFromOrderInd = A, the Customer Reference Number will be defined by the Gateway and any value passed in this element will be ignored.<br><br>- Given that this value can be the same as the Order Number, the valid characters for this field follow the same convention as the Order ID element. Those valid characters are:<br><br>abcdefghijklmnopqrstuvwxyzABCDEFGHIJKLMNOPQRSTUVWXYZ0123456789-,$@& and a space character. However, the space character cannot be the leading character.<br><br>This value may not be changed through a Profile Update action. | C | 22 | A |
| **CustomerProfileOrderOverrideInd** | NewOrder | **Defines if any Order Data can be pre-populated from the Customer Reference Number [CustomerRefNum]:**<br><br>Mandatory if Customer Profile Action Type = Create<br><br>Optional if Customer Profile Action Type = Update<br><br>**NO** No mapping to order data<br><br>**OI** Use <*CustomerRefNum*> for <*OrderID*> and <*ECOrderNum*> or <*MailOrderNum*><br><br>**OD** Use <*CustomerReferNum*> for <*Comments*><br><br>**OA** Use <*CustomerRefNum*> for <*OrderID*> and <*Comments*> | C | 2 | A |

| XML Name | XML Parent Element | Description | Req. M/C/O | Max. Char. | Field Type A/N |
|---|---|---|---|---|---|
| **authenticationECIInd** | NewOrder | **Defines transaction type:**<br>Conditionally required for Verified by Visa and MasterCard SecureCode transactions<br>5 – Verified by Visa/MasterCard SecureCode – Authenticated Transaction<br>6 – Verified by Visa – Attempted Authentication | C | 1 | N |
| **CAVV** | NewOrder | **Cardholder Authentication Verification Value use in Verified by Visa Transactions (CAVV):**<br>- This number must be Base 64 Encoded.<br>- Cryptographic value derived with an algorithm that applies the Issuer's private key to the combination of the Cardholder account number, the Transaction Identifier (XID), and other data. | C | 40 | A |
| **XID** | NewOrder | **Transaction ID used in Verified by Visa Transactions (XID):**<br>▪ This number must be Base 64 Encoded.<br>▪ Unique tracking number set by the Merchant and sent to the Issuer Authentication/Service in the Authentication Request message [Optional] | O | 40 | A |
| **PriorAuthID** | NewOrder | **Defines the transaction type as a Prior Auth:**<br>When this value is present, the request is considered a force authorization. No on-line authorization will be generated to the Host systems. | O | 6 | A |
| **OrderID** | NewOrder | **Merchant Defined Order Number:**<br>Field defined and supplied by the auth originator, and echoed back in response.<br>The first 8 characters should be unique for each transaction.<br>Valid Characters:<br>abcdefghijklmnopqrstuvwxyzABCDEFGHIJKLMNOPQRSTUVWXYZ0123456789-,$@ and a space character. However, the space character cannot be the leading character.<br>PINLess Debit transactions can only use upper and lower case alpha [A-Z, a-z] and numeric [0-9] characters and NO special characters. | M | 22 | A |

| XML Name | XML Parent Element | Description | Req. M/C/O | Max. Char. | Field Type A/N |
|---|---|---|---|---|---|
| **Amount** | NewOrder | **Transaction Amount**:<br>Keys:<br>- Implied Decimal including those currencies that are a zero exponent. For example, both $100.00 (an exponent of '2') and 100 Yen (an exponent of '0') should be sent as <Amount>10000</Amount>. | M | 12 | N |
| **Comments** | NewOrder | **Free form comments:**<br>Merchant can fill in this field and the info will be stored with the transaction details.<br>For PNS customers, this field will populate the Customer Defined Data field, which is displayed in Resource Online. | O | 64 | A |
| **ShippingRef** | NewOrder | **Shipping Tracking Reference Number.**<br>Merchant can fill in this field and the info will be stored with the transaction details. | O | 40 | A |
| **TaxInd** | NewOrder | **Defines the tax type:**<br>Conditionally required for Purchasing Card Level II Data<br>0 – Not provided<br>1 – Included<br>2 – Non-Taxable | C | 1 | N |
| **Tax** | NewOrder | **Tax Amount for the purchase:**<br>- Conditionally required for Purchasing Card Level II Data<br>- Implied decimal including those currencies that are a zero exponent. | C | 12 | N |

| XML Name | XML Parent Element | Description | Req. M/C/O | Max. Char. | Field Type A/N |
|---|---|---|---|---|---|
| **AMEXTranAdvAddn1** | NewOrder | **Amex Purchasing Card Data – Transaction Advice Addendum #1**<br>The TAA Record is used to further identify the purchase that is associated with the charge to the cardholder. It is also used in Purchasing / Procurement card transactions to provide specific details about the transaction to the cardholder for tracking purposes. TAA's should be as concise as possible. A TAA of "Merchandise" for example, would not be acceptable.<br>Salem Only / Conditionally required for Amex Purchasing Card Data | C | 40 | A |
| **AMEXTranAdvAddn2** | NewOrder | **Amex Purchasing Card Data – Transaction Advice Addendum #2**<br>Salem Only / Conditionally required for Amex Purchasing Card Data | C | 40 | A |
| **AMEXTranAdvAddn3** | NewOrder | **Amex Purchasing Card Data – Transaction Advice Addendum #3**<br>Salem Only / Conditionally required for Amex Purchasing Card Data | C | 40 | A |
| **AMEXTranAdvAddn4** | NewOrder | **Amex Purchasing Card Data – Transaction Advice Addendum #4**<br>Salem Only / Conditionally required for Amex Purchasing Card Data | C | 40 | A |
| **AAV** | NewOrder | **Accountholder Authentication Value for MasterCard Secure Code:**<br>Conditionally required for MasterCard SecureCode transactions.<br>This number must be Base 64 Encoded.<br>Unique transaction token generated by the issuer and presented to the merchant each time a cardholder conducts an electronic transaction using MasterCard SecureCode. AAV incorporates elements specific to the transaction and effectively binds the cardholder to a transaction at a merchant for a given sales amount. | C | 32 | A |

| XML Name | XML Parent Element | Description | Req. M/C/O | Max. Char. | Field Type A/N |
|---|---|---|---|---|---|
| **SDMerchantName** | NewOrder | **Soft Descriptor Merchant Name**<br>Conditionally required for Soft Descriptors<br>The Merchant Name field should be what is most recognizable to the cardholder [Company name or trade name]. The actual length of this field is conditionally tied to Host and the Size of the <SDProductDescription> field used.<br>Salem:<br>- CREDIT – Three options, which conditionally affects the SDProductDescription [see below]:<br>o Max 3 bytes<br>o Max 7 bytes<br>o Max 12 bytes<br>- ECP:<br>o Max 15 bytes<br>PNS:<br>- Max 25 bytes. | C | 25 | A |

| XML Name | XML Parent Element | Description | Req. M/C/O | Max. Char. | Field Type A/N |
|---|---|---|---|---|---|
| **SDProductDescription** | NewOrder | **Soft Descriptor Product Description**<br>Conditionally required for Soft Descriptors.<br>Provides an accurate product description<br>Salem:<br>- CREDIT:<br>o If SDMerchantName = 3 bytes – then Max = 18 bytes<br>o If SDMerchantName = 7 bytes – then Max = 14 bytes<br>o If SDMerchantName = 12 bytes – then Max = 9 bytes<br>- ECP:<br>o 10 bytes Max<br>PNS:<br>- This field will **not** show on Cardholder statements for PNS Merchants. | C | 18 | A |
| **SDMerchantCity** | NewOrder | **Soft Descriptor Merchant City**<br>Tag conditionally required for Soft Descriptors.<br>Merchant City for Retail. Field required but should be null filled if any Soft Descriptor data is submitted. | C | 13 | A |

| XML Name | XML Parent Element | Description | Req. M/C/O | Max. Char. | Field Type A/N |
|---|---|---|---|---|---|
| **SDMerchantPhone** | NewOrder | **Soft Descriptor Merchant Phone**<br>Tag conditionally required for Soft Descriptors<br>Only one of the location Soft Descriptor records should be sent [meaning Phone, URL, or Email].<br>This field will **not** show on Cardholder statements for PNS Merchants.<br>Valid Formats:<br>- NNN-NNN-NNNN<br>- NNN-AAAAAAA<br>NOTE: For MasterCard MOTO (Transaction Type 1) and Recurring (Transaction Type 2), if the City/Phone field at the division level is not a Customer Service Phone Number, then a Customer Service Phone Number must be populated in the Merchant city/Customer Phone Number field or the transaction will reject with Response Reason Code BP (Missing Customer Service Phone). | C | 12 | A |
| **SDMerchantURL** | NewOrder | **Soft Descriptor Merchant URL**<br>Tag conditionally required for Soft Descriptors [can be null filled]<br>Only one of the location Soft Descriptor records should include data [meaning Phone, URL, or Email].<br>This field will **not** show on Cardholder statements for PNS Merchants. | C | 13 | A |
| **SDMerchantEmail** | NewOrder | **Soft Descriptor Merchant Email**<br>Tag conditionally required for Soft Descriptors [can be null filled]<br>Only one of the location Soft Descriptor records should include data [meaning Phone, URL, or Email].<br>This field will **not** show on Cardholder statements for PNS Merchants. | C | 13 | A |

| XML Name | XML Parent Element | Description | Req. M/C/O | Max. Char. | Field Type A/N |
|---|---|---|---|---|---|
| **RecurringInd** | NewOrder | **Recurring indicator:**<br>This tag is conditionally required for Merchants that are:<br>- Located in Canada<br>- Processing on BIN 000002<br>- Industry Type = Recurring<br>The valid values are:<br>RF – First Recurring Transaction.<br>RS – Subsequent Recurring Transactions. | C | 2 | A |
| **EUDDCountryCode** | NewOrder | **European Direct Debit Country Code**<br>Customer's Country Code. The following is the list of valid country codes.<br>AT – Austria<br>BE – Belgium<br>DE – Germany<br>FR – France<br>GB – United Kingdom<br>NL – Netherlands | C | 2 | A |
| **EUDDBankSortCode** | NewOrder | **European Direct Debit Bank Sort Code**<br>Customer's Bank Sort code. Mandatory for the following Country Codes:<br>AT – Austria<br>DE – Germany<br>FR – France<br>GB – United Kingdom | C | 10 | A |
| **EUDDRibCode** | NewOrder | **European Direct Debit RIB**<br>Bank Account checksum – used in France only | C | 2 | A |
| **BMLCustomerIP** | NewOrder | **Customer's IP Address**<br>Optional for Bill Me Later sale transactions | O | 45 | A |
| **BMLCustomerEmail** | NewOrder | **Customer Email Address**<br>Optional for Bill Me Later sale transactions | O | 50 | A |

| XML Name | XML Parent Element | Description | Req. M/C/O | Max. Char. | Field Type A/N |
|---|---|---|---|---|---|
| **BMLShippingCost** | NewOrder | **Total Shipping Cost of Consumers Order**<br>Mandatory for Bill Me Later sale transactions | C | 8 | N |
| **BMLTNCVersion** | NewOrder | **Terms and Conditions Number**<br>The Terms and Conditions Number to which the consumer agreed.<br>Mandatory for Bill Me Later sale transactions | C | 5 | N |
| **BMLCustomerRegistrationDate** | NewOrder | **Customer Registration Date**<br>The date a customer registered with the merchant.<br>Mandatory for Bill Me Later sale transactions | C | 8 | N |
| **BMLCustomerTypeFlag** | NewOrder | **Customer Type Flag**<br>New or Existing Customer to the Merchant [not Bill Me Later]<br>Valid Values:<br>N – New<br>E – Existing<br>Optional for Bill Me Later sale transactions | O | 2 | A |
| **BMLItemCategory** | NewOrder | **Item Category**<br>Product Description Code assigned by I4Commerce<br>Mandatory for Bill Me Later sale transactions | C | 4 | N |
| **BMLPreapprovalInvitationNum** | NewOrder | **Pre-Approval Invitation Number**<br>Indicates whether the consumer has been pre-approved for Bill Me Later or not.<br>▪ Pre-approval from credit bureau should include the 16-digit pre-approval number. This will allow the pre-approval to be matched with the first consumer order.<br>▪ Internal pre-approval should include the leftmost digit as a 1.<br>▪ No pre-approval should include all zeros or be blank filled.<br>Optional for Bill Me Later sale transactions | O | 16 | A |

| XML Name | XML Parent Element | Description | Req. M/C/O | Max. Char. | Field Type A/N |
|---|---|---|---|---|---|
| **BMLMerchantPromotionalCode** | NewOrder | **Merchant Promotional Code**<br>Optional for Bill Me Later sale transactions | O | 4 | A |
| **BMLCustomerBirthDate** | NewOrder | **Customer Date of Birth**<br>YYYYMMDD Format<br>Mandatory for Bill Me Later sale transactions | C | 8 | N |
| **BMLCustomerSSN** | NewOrder | **Customer Social Security Number**<br>Either the full 9 digit or last 4 digits of the customer's Social Security Number.<br>Mandatory for Bill Me Later sale transactions | C | 9 | N |
| **BMLCustomerAnnualIncome** | NewOrder | **Gross Household Annual Income**<br>Implied Decimal. For example, $100,000.00 should be sent as:<br><BMLCustomerAnnualIncome>10000000</BMLCustomerAnnualIncome><br>Optional for Bill Me Later sale transactions | O | 10 | N |
| **BMLCustomerResidenceStatus** | NewOrder | **Customer Residence Status**<br>Valid Values:<br>O – Own<br>R – Rent<br>X – Other<br>Optional for Bill Me Later sale transactions | O | 1 | A |
| **BMLCustomerCheckingAccount** | NewOrder | **Customer Checking Account Indicator**<br>Valid Values:<br>Y – Yes, customer has a checking account<br>N – No, customer does not have a checking account<br>Optional for Bill Me Later sale transactions | O | 1 | A |

| XML Name | XML Parent Element | Description | Req. M/C/O | Max. Char. | Field Type A/N |
|---|---|---|---|---|---|
| **BMLCustomerSavingsAccount** | NewOrder | **Customer Savings Account Indicator**<br>Valid Values:<br>Y – Yes, customer has a savings account<br>N – No, customer does not have a savings account<br>Optional for Bill Me Later sale transactions | O | 1 | A |
| **BMLProductDeliveryType** | NewOrder | **Delivery Type Indicator**<br>Valid Values:<br>CNC – Cash and Carry<br>DIG – Digital Goods<br>PHY – Physical Delivery Required<br>SVC – Service<br>TBD – To Be Determined<br>Optional for Bill Me Later sale transactions | C | 3 | A |
| **BillerReferenceNumber** | NewOrder | **Biller Reference Number [PINLess Debit Only]**<br>Reference Number the Biller (merchant) uses on their system to identify this customer.<br>Conditionally required for PINLess Debit | C | 25 | A |
| **PCOrderNum** | NewOrder | **PO number or Order number from customer:**<br>Required for Purchasing Card Level 2 Data | C | 17 | A |
| **PCDestZip** | NewOrder | **Shipping Destination Zip code for the purchase:**<br>- Required for Purchasing Card Level 2 and Level 3 Data<br>- For Zip Code + 4 please separate with '-'.<br>Required for best Interchange rate and cannot be all zeros or nines. | C | 10 | A |
| **PCDestName** | NewOrder | **Amex Purchasing Card Data – Cardholder Ship To: Name**<br>Salem Only / Required for Amex Purchasing Card Data | C | 30 | A |
| **PCDestAddress1** | NewOrder | **Amex Purchasing Card Data - Cardholder Ship To: Address line 1**<br>Salem Only / Required for Amex Purchasing Card Data | C | 30 | A |

| XML Name | XML Parent Element | Description | Req. M/C/O | Max. Char. | Field Type A/N |
|---|---|---|---|---|---|
| **PCDestAddress2** | NewOrder | **Amex Purchasing Card Data - Cardholder Ship To: Address line 2**<br>Salem Only / Required for Amex Purchasing Card Data | C | 30 | A |
| **PCDestCity** | NewOrder | **Amex Purchasing Card Data – Cardholder Ship TO: City**<br>Salem Only / Required for Amex Purchasing Card Data | C | 20 | A |
| **PCDestState** | NewOrder | **Amex Purchasing Card Data – Cardholder Ship TO: State**<br>Salem Only / Required for Amex Purchasing Card Data | C | 2 | A |
| **PC3FreightAmt** | NewOrder | **Purchase Card Level 3 freight amount for shipment**<br>Total freight or shipping and handling charges. Implied decimal. | O | 12 | N |
| **PC3DutyAmt** | NewOrder | **Purchase Card Level 3 Duty Amount for Shipment**<br>Total charges for any import and/or export duties included in this transaction. Implied decimal. | O | 12 | N |
| **PC3DestCountryCd** | NewOrder | **Purchase Card Level 3 Destination Country Code**<br>The ISO-assigned code of the country to which the goods are shipped. Conditionally required for all Purchasing Card 3 transactions. If no value is submitted, it will be default to the United States [USA].<br>Required for Purchasing Card 3.<br>See Table in Appendix A | C | 3 | A |
| **PC3ShipFromZip** | NewOrder | **Purchase Card Level 3 Ship from Zip**<br>The zip/postal code of the location from which the goods are shipped.<br>Required for best Interchange rate and cannot be all zeros or nines. | C | 10 | A |

| XML Name | XML Parent Element | Description | Req. M/C/O | Max. Char. | Field Type A/N |
|---|---|---|---|---|---|
| **PC3DiscAmt** | NewOrder | **Purchase Card Level 3 Discount Amount from Order**<br>The total amount of discount applied to the transaction by the merchant. Used by the merchant when a price break is given on an entire transaction rather than on unit prices. Typically, this is shown as a credit on a detailed invoice.<br>Implied decimal.<br>Optional for Visa only. Should not be sent for MasterCard. | O | 12 | N |
| **PC3VATtaxAmt** | NewOrder | **Purchase Card Level 3 Total Amount of VAT or other tax**<br>The total amount of VAT or other tax included in this transaction.<br>Implied decimal<br>Optional for Visa only. Should not be sent for MasterCard. | O | 12 | N |
| **PC3VATtaxRate** | NewOrder | **Purchase Card Level 3 Rate of VAT or other tax**<br>The total amount of VAT or other tax (expressed in percentage terms) for this line item.<br>2 decimal implied.<br>Example: 0001 = 1%<br>Optional for Visa only. Should not be sent for MasterCard. | O | 4 | N |
| **PC3AltTaxID** | NewOrder | **Purchase Card Level 3 Alternate Tax ID**<br>Tax ID number for the alternate tax associated with this transaction.<br>Optional for MasterCard only. However, required if an amount is sent in PC3AltTaxAmt. Should not be sent for Visa. | O | 15 | N |
| **PC3AltTaxAmt** | NewOrder | **Purchase Card Level 3 Alternate Tax Amount**<br>Total Amount of alternate tax associated with this transaction.<br>Implied decimal.<br>Optional for MasterCard only. However, required if an amount is sent in PC3AltTaxID. Should not be sent for Visa. | O | 9 | N |

| XML Name | XML Parent Element | Description | Req. M/C/O | Max. Char. | Field Type A/N |
|---|---|---|---|---|---|
| **PC3LineItemCount** | NewOrder | **Purchase Card Level 3 Number of Line Items**<br>The number of Purchasing Card 3 Line Item Detail items included with this transaction. The maximum number of line items is 98. At least 1 line item must be included to submit Purchasing Card 3 data.<br>Required for Purchasing Card 3. | C | 2 | N |
| **PC3LineItemArray** | NewOrder | **Purchase Card Level 3 Detail Header**<br>Required parent tag for Purchasing Card 3 Line Item Detail components. | C | N/A | N/A |
| **PC3LineItem** | PC3LineItemArray | **Parent XML Tag for Individual Purchase Card Level 3 Line Item Details**<br>This XML Tag is the parent for each Line item detail included in this transaction. It should be repeated for each item. | C | N/A | N/A |
| **PC3DtlIndex** | PC3LineItem | **Purchase Card Level 3 Line Item Index**<br>The sequential number [1 – 98] of the Line Item Details included with this transaction.<br>Required for Purchasing Card 3. | C | 2 | N |
| **PC3DtlDesc** | PC3LineItem | **Purchase Card level 3 Line Item Detail Element – Description**<br>Text description of the item purchased.<br>Cannot be all zeros<br>Required for Purchasing Card 3. | C | 35 | A |
| **PC3DtlProdCd** | PC3LineItem | **Purchase Card level 3 Line Item Detail Element – Product Code**<br>Product code of the item purchased.<br>Cannot be all zeros<br>Required for Purchasing Card 3. | C | 12 | A |

| XML Name | XML Parent Element | Description | Req. M/C/O | Max. Char. | Field Type A/N |
|---|---|---|---|---|---|
| **PC3DtlQty** | PC3LineItem | **Purchase Card level 3 Line Item Detail Element – Number of Units**<br>Number of units of the item purchased.<br>Cannot be all zeros<br>Required for Purchasing Card 3.<br>Implied decimal of 4. | C | 13 | N |
| **PC3DtlUOM** | PC3LineItem | **Purchase Card level 3 Line Item Detail Element – Unit of Measurement**<br>The unit of measure, or unit of measure code used for this line item.<br>Required for Purchasing Card 3.<br>See Table in Appendix A | C | 3 | A |
| **PC3DtlTaxAmt** | PC3LineItem | **Purchase Card level 3 Line Item Detail Element – Tax Amount**<br>The tax amount for this item<br>Implied decimal<br>Required for Purchasing Card 3 | C | 13 | N |
| **PC3DtlTaxRate** | PC3LineItem | **Purchase Card level 3 Line Item Detail Element – Tax Rate**<br>Tax rate applied for this item.<br>Implied decimal of 3 as a percentage. Ex: Interest rate of 6.25% should be sent as'06250'.<br>Required for Purchasing Card 3 | C | 5 | N |

| XML Name | XML Parent Element | Description | Req. M/C/O | Max. Char. | Field Type A/N |
|---|---|---|---|---|---|
| **PC3Dtllinetot** | PC3LineItem | **Purchase Card level 3 Line Item Detail Element – Line Item Total**<br>For PNS customers:<br>- This field must equal the Unit Cost [PC3DtlUOM] multiplied by the quantity [PC3DtlQty] less any discounts [PC3DtlDisc]. If it does not, then this transaction will receive an error.<br>- Additionally, the sum of all the Line Item totals [i.e., the sum of all these fields] cannot exceed the transaction amount [<Amount>] submitted for this order.<br>Implied decimal<br>Cannot be all zeros for either PNS or Salem.<br>Required for Purchasing Card 3 | C | 13 | N |
| **PC3DtlDisc** | PC3LineItem | **Purchase Card level 3 Line Item Detail Element – Discount Amount for Line Item**<br>Amount of the discount applied to the line item<br>Implied decimal<br>Required for Purchasing Card 3 | C | 13 | N |
| **PC3DtlCommCd** | PC3LineItem | **Purchase Card level 3 Line Item Detail Element – Commodity Code for Line Item**<br>The commodity code used to classify the item purchased.<br>Required for Visa. Should not be sent for MasterCard. | C | 12 | N |
| **PC3DtlUnitCost** | PC3LineItem | **Purchase Card level 3 Line Item Detail Element – Unit Cost of Item Purchased**<br>Unit Cost of the unit purchased<br>Implied decimal of 4.<br>Required for Purchasing Card 3 | C | 13 | N |

| XML Name | XML Parent Element | Description | Req. M/C/O | Max. Char. | Field Type A/N |
|---|---|---|---|---|---|
| **PC3DtlGrossNet** | PC3LineItem | **Purchase Card level 3 Line Item Detail Element – Gross/Net Indicator**<br>Indicates whether tax amount is included in the item amount<br>Valid values:<br>Y = Item amount includes tax amount<br>N = Item amount does not include tax amount<br>Required for Purchasing Card 3 | C | 1 | A |
| **PC3DtlTaxType** | PC3LineItem | **Purchase Card level 3 Line Item Detail Element – Type of Tax Being Applied**<br>Type of tax being applied | O | 4 | A |
| **PC3DtlDiscInd** | PC3LineItem | **Purchase Card level 3 Line Item Detail Element – Discount Indicator**<br>Indicates whether the amount if discounted<br>Valid values:<br>Y = Amount is discounted<br>N = Amount is not discounted<br>If value = Y, and Discount Amount Field [PC3DiscAmt] is Blank or Zero Filled, Chase Paymentech will change this field indicator to 'N' before sending the data.<br>Optional for MasterCard only. Should not be sent for Visa. | O | 1 | A |
| **PC3DtlDebitInd** | PC3LineItem | **Purchase Card level 3 Line Item Detail Element – Item Debit/Credit Indicator**<br>Valid Values:<br>D = Item extended amount is a debit.<br>C = Item extended amount is a credit.<br>Required for Purchasing Card 3 for PNS [BIN 00002] Merchants | C | 1 | A |

# New Order Response Elements

| XML Name | XML Parent Element | Description | Req. M/C/O | Max. Char. | Field Type A/N |
|---|---|---|---|---|---|
| **Response** | N/A | **Required XML Parent Tag** | M | N/A | N/A |
| **NewOrderResp** | Response | **XML tag that defines the transaction as a New Order response** | M | N/A | N/A |
| **IndustryType** | NewOrderResp | **Defines the Industry type of the transaction:**<br>This tag will return null results | M | 2 | A |
| **MessageType** | NewOrderResp | **Defines the transaction New Order Transaction Type:**<br>Echoes the Message Type passed in the request | M | 2 | A |
| **MerchantID** | NewOrderResp | **Gateway merchant account number assigned by Chase Paymentech Solutions:**<br>Echoes the account number passed in the request | M | 12 | N |
| **TerminalID** | NewOrderResp | **Merchant Terminal ID assigned by Chase Paymentech Solutions:**<br>Echoes the Terminal ID passed in the request | M | 3 | N |
| **CardBrand** | NewOrderResp | **Defines the Card Type / Brand for the Transaction:**<br>Echoes the Card Type/Brand passed in the request, except:<br>▪ If no CardBrand was sent in the request [when optional], such as Visa / MasterCard, the specific Card Brand mnemonic is returned.<br>▪ For PINLess Debit transactions, the request Card Brand is DP [which is a generic PINLess mnemonic. However the response Card Brand will be one of the three supported PINLess Debit Card Brands:<br>o NP - NYCE PINless Debit<br>o PP - Pulse PINless Debit<br>o SP - Star PINless Debit | M | 2 | A |

| XML Name | XML Parent Element | Description | Req. M/C/O | Max. Char. | Field Type A/N |
|---|---|---|---|---|---|
| **AccountNum** | NewOrderResp | **Account Number**<br><br>Value is conditionally returned for approved Bill Me Later transactions. Other methods of payment will never return the card number. | M | 19 | N |
| **OrderID** | NewOrderResp | **Merchant Defined Order Number:**<br><br>Echoes the Order Number passed in the request | M | 22 | A |
| **TxRefNum** | NewOrderResp | **Gateway transaction reference number:**<br><br>A unique value for each transaction, which is required to adjust any transaction in the gateway [such as Mark for Capture or void]. | M | 40 | A |
| **TxRefIdx** | NewOrderResp | **Gateway transaction index:**<br><br>Used to identify the unique components of transactions adjusted more than one time. Required on for void transactions, not for Mark for Captures. | M | 4 | A |
| **ProcStatus** | NewOrderResp | **Process Status:**<br><br>The first element that should be checked to determine the result of a request. It is the only element that is returned in all response scenarios. It identifies whether transactions have successfully passed all of the Gateway edit checks.<br><br>0 – Success<br><br>All other values constitute an error condition. See appendix for definition of those error values. | M | 6 | A |
| **ApprovalStatus** | NewOrderResp | **Approval Status:**<br><br>Conditional on Process Status returning a '0' or successful response. If so, approval status identifies the result of the authorization request to the host system.<br><br>0 – Decline<br>1 – Approved<br>2 – Message/System Error | C | 1 | N |

| XML Name | XML Parent Element | Description | Req. M/C/O | Max. Char. | Field Type A/N |
|---|---|---|---|---|---|
| **RespCode** | NewOrderResp | **Response Code:**<br>Normalized authorization response code issued by the host system [Salem / PNS], which identifies an approval ('00') or the reason for a decline or error.<br>See appendix for values. | C | 2 | A |
| **AVSRespCode** | NewOrderResp | **Address verification request response:**<br>- See appendix for values<br>- Conditional on AVS request being sent. | M | 2 | A |
| **CVV2RespCode** | NewOrderResp | **Card verification value request response:**<br>- See appendix for values<br>- Conditional on card verification request being sent | M | 1 | A |
| **AuthCode** | NewOrderResp | **Issuer approval Code:**<br>Unique transactional level code issued by the bank or service establishment for approvals. PINLess Debit transactions could return blanks or N/A | C | 6 | A |
| **RecurringAdviceCd** | NewOrderResp | **Recurring Payment Advice Code [MasterCard Only]**<br>01=New account information available. Obtain new account information<br>02=Try again later. Recycle transaction in 72 hours.<br>03=Do not try to obtain another form of payment. | M | 2 | N |
| **CAVVRespCode** | NewOrderResp | **Response code to VbV Requests** | M | 1 | A |
| **StatusMsg** | NewOrderResp | **Text message associated with ProcStatus value.** | C | Var | A |
| **RespMsg** | NewOrderResp | **Messages associated with RespCode.** | C | 80 | A |
| **HostRespCode** | NewOrderResp | **Actual host response code:**<br>Exact response sent by host authorization system [non-normalized by the gateway]. For those systems that have already coded to the Salem / PNS authorization response values, they are available via this tag. | C | 3 | A |

| XML Name | XML Parent Element | Description | Req. M/C/O | Max. Char. | Field Type A/N |
|---|---|---|---|---|---|
| **HostAVSRespCode** | NewOrderResp | **Actual host address verification response code:**<br>Exact address verification response sent by host authorization system [non-normalized by the gateway]. For those systems that have already coded to the Salem / PNS authorization response values, they are available via this tag. | C | 2 | A |
| **HostCVV2RespCode** | NewOrderResp | **Actual host card verification response code:**<br>Exact card verification response sent by host authorization system [non-normalized by the gateway]. For those systems that have already coded to the Salem / PNS authorization response values, they are available still via this tag. | C | 1 | A |
| **CustomerRefNum** | NewOrderResp | **The Customer Reference Number**<br>If Customer Profile Action Type = Create and<br>If CustomerProfileFromOrderInd = S, this field will echo the Customer Reference Number sent in the Profile Request. | M | 22 | A |
| **CustomerName** | NewOrderResp | **Customer Billing Name**<br>Value from the Request Returned. | M | 30 | A |
| **ProfileProcStatus** | NewOrderResp | **Result Status of Profile Management:**<br>Communicates the success or failure of a Profile Management request.<br>0 – Success<br>>0 – An error condition, see Appendix A for definition of the specific Profile Management error values. | M | 6 | A |
| **CustomerProfileMessage** | NewOrderResp | **Verbose Text Description associated with ProfileProcStatus** | M | Var | A |
| **BillerReferenceNumber** | NewOrderResp | **Biller Reference Number [PINLess Debit Only]**<br>Echoed for Request | C | 25 | A |
| **RespTime** | NewOrderResp | **Time the transaction was processed by gateway:**<br>Format – 'hh24miss' | M | 6 | N |

# Mark for Capture (of a previous authorization) Request Elements

Optional data should only be sent with the transaction to enhance the original authorization when either the data was not sent in the original authorization or has changed from the original authorization. Please refer to page 8 for further information on this subject.

When a split shipment is marked for capture, the split amount for the shipment should be included in the amount element for this message. The split transaction also results in the creation of a new order for the balance left over from the original authorization and adjustments as required to the original authorization. Upon marking a portion or the remainder of the split transaction, the system will automatically attempt to obtain a new authorization for the new order when it is settled.

| XML Name | XML Parent Element | Description | Req. M/C/O | Max. Char. | Field Type A/N |
|---|---|---|---|---|---|
| **Request** | N/A | **Required XML Parent Tag** | M | N/A | N/A |
| **markForCapture** | Request | **Required XML Parent Tag** | M | N/A | N/A |
| **OrderID** | markForCapture | **Merchant Defined Order Number:**<br>Should use the same OrderID as the original request | M | 22 | A |
| **Amount** | markForCapture | **Transaction Amount**:<br>Keys:<br>- Amount being captured. It can be less than or equal to the original authorization. Anything less than will create a split transaction.<br>- Implied decimal including those currencies that are a zero exponent. For example, both $100.00 (an exponent of '2') and 100 Yen (an exponent of '0') should be sent as <Amount>10000</Amount>.<br>- See table for min/max amount for each currency type. | M | 12 | N |
| **TaxInd** | markForCapture | **Defines the tax type:**<br>Required for Purchasing Card Level II Data<br>0 – Not provided<br>1 – Included<br>2 – Non-Taxable – Not valid for Visa Purchasing Card II qualification | O | 1 | N |

| XML Name | XML Parent Element | Description | Req. M/C/O | Max. Char. | Field Type A/N |
|---|---|---|---|---|---|
| **Tax** | markForCapture | **Tax Amount for the purchase:**<br>- Required for Purchasing Card Level II Data<br>- Implied decimal including those currencies that are a zero exponent. | O | 12 | N |
| **BIN** | markForCapture | **Transaction Routing Definition:**<br>Assigned by Chase Paymentech<br>000001 – Salem<br>000002 – PNS | M | 6 | N |
| **MerchantID** | markForCapture | **Gateway merchant account number assigned by Chase Paymentech:**<br>This account number will match that of your host platform<br>BIN 000001 – 6 digit Salem Division Number<br>BIN 000002 – 12 digit PNS Merchant ID | M | 15 | N |
| **TerminalID** | markForCapture | **Merchant Terminal ID assigned by Chase Paymentech:**<br>All Terminal IDs at present are '001' | M | 3 | N |
| **TxRefNum** | markForCapture | **Gateway transaction reference number:**<br>A unique value for each transaction, which is required to adjust any transaction in the gateway [such as Mark for Capture or void]. | M | 40 | A |
| **PCOrderNum** | markForCapture | **PO number or Order number from customer:**<br>Required for Purchasing Card Level II Data | O | 17 | A |
| **PCDestZip** | markForCapture | **Shipping Destination Zip code for the purchase:**<br>- Required for Purchasing Card Level II Data<br>- For Zip Code + 4 please separate with '-'. | O | 10 | A |
| **PCDestName** | markForCapture | **Amex Purchasing Card Data –**<br>Salem Only / Required for Amex Purchasing Card Data | O | 30 | A |

| XML Name | XML Parent Element | Description | Req. M/C/O | Max. Char. | Field Type A/N |
|---|---|---|---|---|---|
| **PCDestAddress1** | markForCapture | **Amex Purchasing Card Data - Cardholder Ship To: Address line 1**<br>Salem Only / Required for Amex Purchasing Card Data | O | 30 | A |
| **PCDestAddress2** | markForCapture | **Amex Purchasing Card Data - Cardholder Ship To: Address line 2**<br>Salem Only / Required for Amex Purchasing Card Data | O | 30 | A |
| **PCDestCity** | markForCapture | **Amex Purchasing Card Data – Cardholder Ship TO: City**<br>Salem Only / Required for Amex Purchasing Card Data | O | 20 | A |
| **PCDestState** | markForCapture | **Amex Purchasing Card Data – Cardholder Ship TO: State**<br>Salem Only / Required for Amex Purchasing Card Data | O | 2 | A |
| **AMEXTranAdvAddn1** | markForCapture | **Amex Purchasing Card Data – Transaction Advice Addendum #1**<br>The TAA Record is used to further identify the purchase that is associated with the charge to the cardholder. It is also used in Purchasing / Procurement card transactions to provide specific details about the transaction to the cardholder for tracking purposes. TAA's should be as concise as possible. A TAA of "Merchandise" for example, would not be acceptable.<br>Salem Only / Required for Amex Purchasing Card Data | O | 40 | A |
| **AMEXTranAdvAddn2** | markForCapture | **Amex Purchasing Card Data – Transaction Advice Addendum #2**<br>Salem Only / Required for Amex Purchasing Card Data | O | 40 | A |
| **AMEXTranAdvAddn3** | markForCapture | **Amex Purchasing Card Data – Transaction Advice Addendum #3**<br>Salem Only / Required for Amex Purchasing Card Data | O | 40 | A |
| **AMEXTranAdvAddn4** | markForCapture | **Amex Purchasing Card Data – Transaction Advice Addendum #4**<br>Salem Only / Required for Amex Purchasing Card Data | O | 40 | A |

| XML Name | XML Parent Element | Description | Req. M/C/O | Max. Char. | Field Type A/N |
|---|---|---|---|---|---|
| **PC3FreightAmt** | markForCapture | **Purchase Card Level 3 freight amount for shipment**<br>Total freight or shipping and handling charges. Implied decimal. | O | 12 | N |
| **PC3DutyAmt** | markForCapture | **Purchase Card Level 3 Duty Amount for Shipment**<br>Total charges for any import and/or export duties included in this transaction. Implied decimal. | O | 12 | N |
| **PC3DestCountryCd** | markForCapture | **Purchase Card Level 3 Destination Country Code**<br>The ISO-assigned code of the country to which the goods are shipped. Conditionally required for all Purchasing Card 3 transactions. If no value is submitted, it will be default to the United States [USA].<br>Required for Purchasing Card 3.<br>See Table in Appendix A | C | 3 | A |
| **PC3ShipFromZip** | markForCapture | **Purchase Card Level 3 Ship from Zip**<br>The zip/postal code of the location from which the goods are shipped.<br>Required for best Interchange rate and cannot be all zeros or nines. | C | 10 | A |
| **PC3DiscAmt** | markForCapture | **Purchase Card Level 3 Discount Amount from Order**<br>The total amount of discount applied to the transaction by the merchant. Used by the merchant when a price break is given on an entire transaction rather than on unit prices. Typically, this is shown as a credit on a detailed invoice.<br>Implied decimal.<br>Optional for Visa only. Should not be sent for MasterCard. | O | 12 | N |

| XML Name | XML Parent Element | Description | Req. M/C/O | Max. Char. | Field Type A/N |
|---|---|---|---|---|---|
| **PC3VATtaxAmt** | markForCapture | **Purchase Card Level 3 Total Amount of VAT or other tax**<br>The total amount of VAT or other tax (expressed in percentage terms) for this line item.<br>2 decimal implied.<br>Example: 0001 = 1%<br>Optional for Visa only. Should not be sent for MasterCard. | O | 12 | N |
| **PC3VATtaxRate** | markForCapture | **Purchase Card Level 3 Rate of VAT or other tax**<br>The total amount of VAT or other tax included in this transaction.<br>Implied decimal<br>Optional for Visa only. Should not be sent for MasterCard. | O | 4 | N |
| **PC3AltTaxID** | markForCapture | **Purchase Card Level 3 Alternate Tax ID**<br>Tax ID number for the alternate tax associated with this transaction.<br>Optional for MasterCard only. However, required if an amount is sent in PC3AltTaxAmt. Should not be sent for Visa. | O | 15 | N |
| **PC3AltTaxAmt** | markForCapture | **Purchase Card Level 3 Alternate Tax Amount**<br>Total Amount of alternate tax associated with this transaction.<br>Implied decimal.<br>Optional for MasterCard only. However, required if an amount is sent in PC3AltTaxID. Should not be sent for Visa. | O | 9 | N |
| **PC3LineItemCount** | markForCapture | **Purchase Card Level 3 Number of Line Items**<br>The number of Purchasing Card 3 Line Item Detail items included with this transaction. The maximum number of line items is 98. At least 1 line item must be included to submit Purchasing Card 3 data.<br>Required for Purchasing Card 3. | C | 2 | N |

| XML Name | XML Parent Element | Description | Req. M/C/O | Max. Char. | Field Type A/N |
|---|---|---|---|---|---|
| **PC3LineItemArray** | markForCapture | **Purchase Card Level 3 Detail Header**<br>Required parent tag for Purchasing Card 3 Line Item Detail components. | C | N/A | N/A |
| **PC3LineItem** | PC3LineItemArray | **Parent XML Tag for Individual Purchase Card Level 3 Line Item Details**<br>This XML Tag is the parent for each Line item detail included in this transaction. It should be repeated for each item. | C | N/A | N/A |
| **PC3DtlIndex** | PC3LineItem | **Purchase Card Level 3 Line Item Index**<br>The sequential number [1 – 98] of the Line Item Details included with this transaction.<br>Required for Purchasing Card 3. | C | 2 | N |
| **PC3DtlDesc** | PC3LineItem | **Purchase Card level 3 Line Item Detail Element – Description**<br>Text description of the item purchased.<br>Cannot be all zeros<br>Required for Purchasing Card 3. | C | 35 | A |
| **PC3DtlProdCd** | PC3LineItem | **Purchase Card level 3 Line Item Detail Element – Product Code**<br>Product code of the item purchased.<br>Cannot be all zeros<br>Required for Purchasing Card 3. | C | 12 | A |
| **PC3DtlQty** | PC3LineItem | **Purchase Card level 3 Line Item Detail Element – Number of Units**<br>Number of units of the item purchased.<br>Cannot be all zeros<br>Required for Purchasing Card 3.<br>Implied decimal of 4. | C | 13 | N |

| XML Name | XML Parent Element | Description | Req. M/C/O | Max. Char. | Field Type A/N |
|---|---|---|---|---|---|
| **PC3DtlUOM** | PC3LineItem | **Purchase Card level 3 Line Item Detail Element – Unit of Measurement**<br>The unit of measure, or unit of measure code used for this line item.<br>Required for Purchasing Card 3.<br>See Table in Appendix A | C | 3 | A |
| **PC3DtlTaxAmt** | PC3LineItem | **Purchase Card level 3 Line Item Detail Element – Tax Amount**<br>The tax amount for this item<br>Implied decimal<br>Required for Purchasing Card 3 | C | 13 | N |
| **PC3DtlDebitInd** | PC3LineItem | **Purchase Card level 3 Line Item Detail Element – Item Debit/Credit Indicator**<br>Valid Values:<br>D = Item extended amount is a debit.<br>C = Item extended amount is a credit.<br>Required for Purchasing Card 3 for PNS [BIN 00002] Merchants | C | 1 | A |
| **PC3DtlTaxRate** | PC3LineItem | **Purchase Card level 3 Line Item Detail Element – Tax Rate**<br>Tax rate applied for this item.<br>Implied decimal of 3 as a percentage. Ex: Interest rate of 6.25% should be sent as'06250'.<br>Required for Purchasing Card 3 | C | 5 | N |

| XML Name | XML Parent Element | Description | Req. M/C/O | Max. Char. | Field Type A/N |
|---|---|---|---|---|---|
| **PC3Dtllinetot** | PC3LineItem | **Purchase Card level 3 Line Item Detail Element – Line Item Total**<br>For PNS customers:<br>- This field must equal the Unit Cost [PC3DtlUOM] multiplied by the quantity [PC3DtlQty] less any discounts [PC3DtlDisc]. If it does not, then this transaction will receive an error.<br>- Additionally, the sum of all the Line Item totals [i.e., the sum of all these fields] cannot exceed the transaction amount [<Amount>] submitted for this order.<br>Implied decimal<br>Cannot be all zeros<br>Required for Purchasing Card 3 | C | 13 | N |
| **PC3DtlDisc** | PC3LineItem | **Purchase Card level 3 Line Item Detail Element – Discount Amount for Line Item**<br>Amount of the discount applied to the line item<br>Implied decimal<br>Required for Purchasing Card 3 | C | 13 | N |
| **PC3DtlCommCd** | PC3LineItem | **Purchase Card level 3 Line Item Detail Element – Commodity Code for Line Item**<br>The commodity code used to classify the item purchased.<br>Required for Visa. Should not be sent for MasterCard. | C | 12 | N |
| **PC3DtlUnitCost** | PC3LineItem | **Purchase Card level 3 Line Item Detail Element – Unit Cost of Item Purchased**<br>Unit Cost of the unit purchased<br>Implied decimal of 4.<br>Required for Purchasing Card 3 | C | 13 | N |

| XML Name | XML Parent Element | Description | Req. M/C/O | Max. Char. | Field Type A/N |
|---|---|---|---|---|---|
| **PC3DtlGrossNet** | PC3LineItem | **Purchase Card level 3 Line Item Detail Element – Gross/Net Indicator**<br>Indicates whether tax amount is included in the item amount<br>Valid values:<br>Y = Item amount includes tax amount<br>N = Item amount does not include tax amount<br>Required for Purchasing Card 3 | C | 1 | A |
| **PC3DtlTaxType** | PC3LineItem | **Purchase Card level 3 Line Item Detail Element – Type of Tax Being Applied**<br>Type of tax being applied | O | 4 | A |
| **PC3DtlDiscInd** | PC3LineItem | **Purchase Card level 3 Line Item Detail Element – Discount Indicator**<br>Indicates whether the amount if discounted<br>Valid values:<br>Y = Amount is discounted<br>N = Amount is not discounted<br>If value = Y, and Discount Amount Field [PC3DiscAmt] is Blank or Zero Filled, Chase Paymentech will change this field indicator to 'N' before sending the data.<br>Optional for MasterCard only. Should not be sent for Visa. | O | 1 | A |

# Mark for Capture Response Elements

| XML Name | XML Parent Element | Description | Req. M/C/O | Max. Char. | Field Type A/N |
|---|---|---|---|---|---|
| **Response** | N/A | **Required XML Parent Tag** | M | N/A | N/A |
| **MarkForCaptureResp** | Response | **XML tag that defines the transaction as a New Order response** | M | N/A | N/A |
| **MerchantID** | MarkForCaptureResp | **Gateway merchant account number assigned by Chase Paymentech Solutions:**<br>Echoes the account number passed in the request | M | 12 | N |
| **TerminalID** | MarkForCaptureResp | **Merchant Terminal ID assigned by Chase Paymentech Solutions:**<br>Echoes the Terminal ID passed in the request | M | 3 | N |
| **OrderID** | MarkForCaptureResp | **Merchant Defined Order Number:**<br>Echoes the Order Number passed in the request | M | 22 | A |
| **TxRefNum** | MarkForCaptureResp | **Gateway transaction reference number:**<br>Echoes the Transaction Reference Number passed in the request | M | 40 | A |
| **TxRefIdx** | MarkForCaptureResp | **Gateway transaction index:**<br>Used to identify the unique components of transactions adjusted more than one time. Required on for void transactions, not for Mark for Captures. | C | 4 | A |
| **Amount** | MarkForCaptureResp | **Transaction Amount**:<br>Echoes the Amount passed in the request | M | 12 | N |

| XML Name | XML Parent Element | Description | Req. M/C/O | Max. Char. | Field Type A/N |
|---|---|---|---|---|---|
| **ProcStatus** | MarkForCaptureResp | **Process Status:**<br>The first element that should be checked to determine the result of a request. It is the only element that is returned in all response scenarios. It identifies whether transactions have successfully passed all of the Gateway edit checks.<br>0 – Success<br>All other values constitute an error condition. See appendix for definition of those error values. | M | 6 | A |
| **StatusMsg** | MarkForCaptureResp | **Text message associated with ProcStatus value.** | C | Var | A |
| **RespTime** | MarkForCaptureResp | **Time the transaction was processed by gateway:**<br>Format – 'hh24miss' | M | 6 | N |

# Void (full and partial) Request Elements

| XML Name | XML Parent Element | Description | Req. M/C/O | Max. Char. | Field Type A/N |
|---|---|---|---|---|---|
| **Request** | N/A | **Required XML Parent Tag** | M | N/A | N/A |
| **Reversal** | Request | **Required XML Parent Tag** | M | N/A | N/A |
| **TxRefNum** | Reversal | **Gateway transaction reference number:**<br>A unique value for each transaction, which is required to adjust any transaction in the gateway [such as Mark for Capture or void]. | M | 40 | A |
| **TxRefIdx** | Reversal | **Gateway transaction index:**<br>Used to identify the unique components of transactions adjusted more than one time. Required on for void transactions, not for Mark for Captures.<br>- Void of Previous Txn - TxRefIdx = Null<br>- Void of a specific Txn – TxRefIdx = Value returned in response | M | 4 | N |
| **AdjustedAmt** | Reversal | **Transaction Amount**:<br>Keys:<br>- When a specific amount is included with this tag, that amount will be voided [assuming that the amount is not greater than the transaction amount remaining]. However, the absence of this tag on a void transaction will perform a full Void.<br>- Implied decimal including those currencies that are a zero exponent. For example, both $100.00 (an exponent of '2') and 100 Yen (an exponent of '0') should be sent as <Amount>10000</Amount>. | M | 12 | N |
| **OrderID** | Reversal | **Merchant Defined Order Number:**<br>Should use the same OrderID as the original request | M | 22 | A |

| XML Name | XML Parent Element | Description | Req. M/C/O | Max. Char. | Field Type A/N |
|---|---|---|---|---|---|
| **BIN** | Reversal | **Transaction Routing Definition:**<br>Assigned by Chase Paymentech Solutions<br>000001 – Salem | M | 6 | N |
| **MerchantID** | Reversal | **Gateway merchant account number assigned by Chase Paymentech Solutions:**<br>This account number will match that of your host platform<br>BIN 000001 – 6 digit Salem Division Number | M | 15 | N |
| **TerminalID** | Reversal | **Merchant Terminal ID assigned by Chase Paymentech Solutions:**<br>All Terminal IDs at present are '001' | M | 3 | N |

# Void (full and partial) Response Elements

| XML Name | XML Parent Element | Description | Req. M/C/O | Max. Char. | Field Type A/N |
|---|---|---|---|---|---|
| **Response** | N/A | **Required XML Parent Tag** | M | N/A | N/A |
| **ReversalResp** | Response | **XML tag that defines the transaction as a New Order response** | M | N/A | N/A |
| **MerchantID** | ReversalResp | **Gateway merchant account number assigned by Chase Paymentech Solutions:**<br>Echoes the account number passed in the request | M | 12 | N |
| **TerminalID** | ReversalResp | **Merchant Terminal ID assigned by Chase Paymentech Solutions:**<br>Echoes the Terminal ID passed in the request | M | 3 | N |
| **OrderID** | ReversalResp | **Merchant Defined Order Number:**<br>Echoes the Order Number passed in the request | M | 22 | A |
| **TxRefNum** | ReversalResp | **Gateway transaction reference number:**<br>Echoes the Transaction Reference Number passed in the request | M | 40 | A |
| **TxRefIdx** | ReversalResp | **Gateway transaction index:**<br>Used to identify the unique components of transactions adjusted more than one time. Required on for void transactions, not for Mark for Captures. | C | 4 | A |
| **OutstandingAmt** | ReversalResp | **Remaining Non-voided amount for partial Voids** | M | 12 | N |
| **ProcStatus** | ReversalResp | **Process Status:**<br>The first element that should be checked to determine the result of a request. It is the only element that is returned in all response scenarios. It identifies whether transactions have successfully passed all of the Gateway edit checks.<br>0 – Success<br>All other values constitute an error condition. See appendix for definition of those error values. | M | 6 | A |

| XML Name | XML Parent Element | Description | Req. M/C/O | Max. Char. | Field Type A/N |
|---|---|---|---|---|---|
| **StatusMsg** | ReversalResp | **Text message associated with ProcStatus value.** | C | Var | A |
| **RespTime** | ReversalResp | **Time the transaction was processed by gateway:**<br>Format – 'hh24miss' | M | 6 | N |

# End of Day Request Elements

This transaction type will cause all Marked for Capture Transactions [Sales and Refunds] to be submitted for Clearing (Batch Request) Elements. From an "End of Day" perspective, a merchant may be configured in one of two ways on the Chase Paymentech Internet Gateway: Auto-settle or Merchant initiated settlement. Merchant's configured in auto-settle mode can ignore the message types defined in this section.

| XML Label | XML Parent Tag | Description | Req. M/C/O | Max. Char. | Field Type A/N |
|---|---|---|---|---|---|
| **Request** | N/A | **Required XML Parent Tag** | M | N/A | N/A |
| **endOfDay** | Request | **XML tag that defines the transaction as an Batch request** | M | N/A | N/A |
| **BIN** | endOfDay | **Transaction Routing Definition:**<br>Assigned by Chase Paymentech<br>000001 – Salem<br>000002 – PNS | M | 6 | N |
| **MerchantID** | endOfDay | **Gateway merchant account number assigned by Chase Paymentech:**<br>This account number will match that of your host platform<br>BIN 000001 – 6 digit Salem Division Number<br>BIN 000002 – 12 digit PNS Merchant ID | M | 15 | N |
| **TerminalID** | endOfDay | **Merchant Terminal ID assigned by Chase Paymentech:**<br>All Terminal IDs at present are '001' | M | 3 | N |

# End of Day Response Elements

| XML Name | XML Parent Element | Description | Req. M/C/O | Max. Char. | Field Type A/N |
|---|---|---|---|---|---|
| **Response** | N/A | **Required XML Parent Tag** | M | N/A | N/A |
| **EndOfDayResp** | Response | **XML tag that defines the transaction as a New Order response** | M | N/A | N/A |
| **MerchantID** | EndOfDayResp | **Gateway merchant account number assigned by Chase Paymentech Solutions:**<br>Echoes the account number passed in the request | M | 12 | N |
| **TerminalID** | EndOfDayResp | **Merchant Terminal ID assigned by Chase Paymentech Solutions:**<br>Echoes the Terminal ID passed in the request | M | 3 | N |
| **BatchSeqNum** | EndOfDayResp | **Sequence Number that References a Settlement Batch** | M | 32 | A |
| **ProcStatus** | EndOfDayResp | **Process Status:**<br>The first element that should be checked to determine the result of a request. It is the only element that is returned in all response scenarios. It identifies whether transactions have successfully passed all of the Gateway edit checks.<br>0 – Success<br>All other values constitute an error condition. See appendix for definition of those error values. | M | 6 | A |
| **StatusMsg** | EndOfDayResp | **Text message associated with ProcStatus value.** | C | Var | A |
| **RespTime** | EndOfDayResp | **Time the transaction was processed by gateway:**<br>Format – 'hh24miss' | M | 6 | N |

# Profile Request Elements

This request type is explicitly for managing Profiles and not using them; including Profile Adds, Updates, Deletes and Inquiries.

| XML Label | XML Parent Tag | Description | Req. M/C/O | Max. Char. | Field Type A/N |
|---|---|---|---|---|---|
| **Request** | N/A | **Required XML Parent Tag** | M | N/A | N/A |
| **Profile** | Request | **XML tag that defines the transaction as an Batch request** | M | N/A | N/A |
| **CustomerBin** | Profile | **Transaction Routing Definition:**<br>Assigned by Chase Paymentech<br>000001 – Salem<br>000002 – PNS<br>This value may not be changed through a Profile Update action.<br>*This is the equivalent to the <BIN> element used on transactional requests* | M | 6 | N |
| **CustomerMerchantID** | Profile | **Gateway merchant account number assigned by Chase Paymentech:**<br>This account number will match that of your host platform<br>BIN 000001 – 6 digit Salem Division Number<br>BIN 000002 – 12 digit PNS Merchant ID<br>This value may not be changed through a Profile Update action.<br>*This is the equivalent to the <MerchantID> element used on transactional requests* | M | 15 | N |
| **CustomerName** | Profile | **Customer Billing Name**<br>Conditionally required for electronic check Profiles<br>*This is the equivalent to the <AVSname> element used on transactional requests.* | C | 30 | A |

| XML Label | XML Parent Tag | Description | Req. M/C/O | Max. Char. | Field Type A/N |
|---|---|---|---|---|---|
| **CustomerRefNum** | Profile | **Sets the Customer Reference Number that will be used to utilize a Customer Profile on all future Orders.**<br><br>Mandatory if Customer Profile Action Type = Read, Update, or Delete<br><br>Or<br><br>Create and the Customer Profile Number generation option = S [Use CustomerRefNum Element]<br><br>Keys:<br><br>- If CustomerProfileFromOrderInd = A, the Customer Reference Number will be defined by the Gateway and any value passed in this element will be ignored.<br><br>- Given that this value can be the same as the Order Number, the valid characters for this field follow the same convention as the Order ID element. Those valid characters are:<br><br>abcdefghijklmnopqrstuvwxyzABCDEFGHIJKLMNOPQRSTUVWXYZ0123456789-,$@& and a space character. However, the space character cannot be the leading character.<br><br>This value may not be changed through a Profile Update action. | C | 22 | A |
| **CustomerAddress1** | Profile | **Cardholder Billing Address line 1**<br><br>Optional if Customer Profile Action Type = Create or Update<br><br>*This is the equivalent to the <AVSaddress1> element used on transactional requests.* | O | 30 | A |
| **CustomerAddress2** | Profile | **Cardholder Billing Address line 2**<br><br>Optional if Customer Profile Action Type = Create or Update<br><br>*This is the equivalent to the <AVSaddress2> element used on transactional requests.* | O | 30 | A |

| XML Label | XML Parent Tag | Description | Req. M/C/O | Max. Char. | Field Type A/N |
|---|---|---|---|---|---|
| **CustomerCity** | Profile | **Cardholder Billing City**<br>Optional if Customer Profile Action Type = Create or Update<br>*This is the equivalent to the <AVScity> element used on transactional requests.* | O | 20 | A |
| **CustomerState** | Profile | **Cardholder Billing State**<br>This is the equivalent to the <AVSstate> element used on transactional requests.<br>Optional if Customer Profile Action Type = Create, or Update | O | 2 | A |
| **CustomerZIP** | Profile | **Cardholder Billing Address Zip Code:**<br>- All AVS requests must minimally include the 5-digit Zip Code.<br>- If sending Zip Code + 4, please separate with a '-'<br>Conditionally required if Customer Profile Action Type = Create<br>*This is the equivalent to the <AVSzip> element used on transactional requests.* | C | 10 | A |
| **CustomerEmail** | Profile | **Cardholder Email**<br>Optional if Customer Profile Action Type = Create or Update<br>*There is no equivalent to this element available on transactional requests.* | O | 50 | A |

| XML Label | XML Parent Tag | Description | Req. M/C/O | Max. Char. | Field Type A/N |
|---|---|---|---|---|---|
| **CustomerPhone** | Profile | **Cardholder Telephone Number:**<br>Format = AAAEEENNNNXXXX, where<br>AAA – Area Code<br>EEE – Exchange<br>NNNN – Number<br>XXXX – EXTENSION<br>**OPTIONAL IF CUSTOMER PROFILE ACTION TYPE = CREATE OR UPDATE**<br>***THIS IS THE EQUIVALENT TO THE <AVSPHONENUM> ELEMENT USED ON TRANSACTIONAL REQUESTS.*** | O | 14 | A |
| **CustomerProfileAction** | Profile | **Defines the Customer Profile action desired:**<br>C – Create a Customer Profile<br>U – Update a Customer Profile<br>R – Retrieve a Customer Profile's Attributes<br>D – Delete a Customer Profile<br>*This element is only used for Customer Profile Management actions.* | M | 6 | A |
| **CustomerProfileOrderOverrideInd** | Profile | **Defines if any Order Data can be pre-populated from the Customer Reference Number [CustomerRefNum]:**<br>Mandatory if Customer Profile Action Type = Create<br>Optional if Customer Profile Action Type = Update<br>**NO** No mapping to order data<br>**OI** Use <*CustomerRefNum*> for <*OrderID*> and <*ECOrderNum*> or <*MailOrderNum*><br>**OD** Use <*CustomerReferNum*> for <*Comments*><br>**OA** Use <*CustomerRefNum*> for <*OrderID*> and <*ECOrderNum*> [or <*MailOrderNum*>] and <*Comments*> | C | 2 | A |

| XML Label | XML Parent Tag | Description | Req. M/C/O | Max. Char. | Field Type A/N |
|---|---|---|---|---|---|
| **CustomerProfileFromOrderInd** | Profile | **Customer Profile Number generation Options:**<br>A – Auto Generate the CustomerRefNum<br>S – Use CustomerRefNum Element<br>When performing an action other than a Customer Profile Action Type = Create, insert "empty" as it will be ignored. But this Attribute is mandatory and cannot be null filled. | M | 5 | A |
| **OrderDefaultDescription** | Profile | **Order Description**<br>Optional if Customer Profile Action Type = Create or Update<br>If the CustomerProfileOrderOverideInd = OA, do not set this value since the Customer Reference Number has been defaulted as the Order Description.<br>*This is the equivalent to the <Comments> element used on transactional requests.* | O | 64 | A |
| **OrderDefaultAmount** | Profile | **Transaction Amount**:<br>Optional if Customer Profile Action Type = Create or Update<br>Keys:<br>- Implied decimal including those currencies that are a zero exponent. For example, both $100.00 (an exp. of '2') and ¥100 (an exp. of '0') should be sent as <OrderDefaultAmount>10000</OrderDefaultAmount><br>- Given that each Orbital Gateway Merchant ID is restricted to one currency, the Currency Code [and Exponent] is defaulted based on Merchant ID in which a transaction presented.<br>*This is the equivalent to the <Amount> element used on transactional requests.* | O | 12 | N |

| XML Label | XML Parent Tag | Description | Req. M/C/O | Max. Char. | Field Type A/N |
|---|---|---|---|---|---|
| **CustomerAccountType** | Profile | **Defines the Customers Payment Type:**<br><br>C – Credit Card<br>S – Switch / Solo<br>E – Electronic Check<br><br>Mandatory if Customer Profile Action Type = Create<br><br>Optional if Customer Profile Action Type = Update<br><br>*This is the equivalent to the multiple XML elements used to define cardholder on transactional requests.* | C | 1 | A |
| **CCAccountNum** | Profile | **Customer Credit Card Number:**<br><br>Mandatory if Customer Profile Action Type = Create and the Customer Payment Type = Credit card or Switch / Solo<br><br>Optional if Customer Profile Action Type = Update and the Customer Payment Type = Credit card or Switch / Solo<br><br>*This is the equivalent to the <AccountNum> element used on transactional requests.* | C | 19 | N |
| **CCExpireDate** | Profile | **Customer Credit Card Expiration Date:**<br><br>Mandatory if Customer Profile Action Type = Create and the Customer Payment Type = Credit card or Switch / Solo<br><br>Optional if Customer Profile Action Type = Update and the Customer Payment Type = Credit card or Switch / Solo<br><br>- Formats: MMYY<br><br>Salem {CustomerBIN 000001] allows a blank to be submitted when no known EXP date exists. Please discuss this feature with your certification analyst before implementing. There are three valid mechanisms for submitting a 'Blank' expiration date using Orbital to the Salem Host. They are:<br><br>- Null fill this XML element - <Exp/><br>- Send four spaces - <Exp> </Exp><br>- Zero fill the XML Element - <Exp>0000</Exp><br><br>*This is the equivalent to the <Exp> element used on transactional requests.* | C | 4 | N |

| XML Label | XML Parent Tag | Description | Req. M/C/O | Max. Char. | Field Type A/N |
|---|---|---|---|---|---|
| **ECPAccountDDA** | Profile | **ECP [DDA] Account Number:**<br>Mandatory if Customer Profile Action Type = Create and the Customer Payment Type = ECP<br>Optional if Customer Profile Action Type = Update and the Customer Payment Type = ECP<br>*This is the equivalent to the <CheckDDA> element used on transactional requests.* | C | 17 | A |
| **ECPAccountType** | Profile | **Defines the ECP deposit account type:**<br>Mandatory if Customer Profile Action Type = Create and the Customer Payment Type = ECP<br>Optional if Customer Profile Action Type = Update and the Customer Payment Type = ECP<br>C – Consumer Checking [US or Canadian]<br>S – Consumer Savings [US Only]<br>X – Commercial Checking [US Only]<br>*This is the equivalent to the <BankAccountType> element used on transactional requests.* | C | 1 | A |
| **ECPAccountRT** | Profile | **ECP Bank routing and transit number for the customer.**<br>Mandatory if Customer Profile Action Type = Create and the Customer Payment Type = ECP<br>Optional if Customer Profile Action Type = Update and the Customer Payment Type = ECP<br>NOTE:<br>- All US Bank Routing Numbers are 9 digits<br>- All Canadian Bank Routing Numbers are 8 Digits<br>This is the equivalent to the <BCRtNum> element used on transactional requests. | C | 9 | N |

| XML Label | XML Parent Tag | Description | Req. M/C/O | Max. Char. | Field Type A/N |
|---|---|---|---|---|---|
| **ECPBankPmtDlv** | Profile | **Defines the ECP payment delivery method:**<br><br>Mandatory if Customer Profile Action Type = Create and the Customer Payment Type = ECP<br><br>Optional if Customer Profile Action Type = Update and the Customer Payment Type = ECP<br><br>This field indicates the preferred manner to deposit the transaction.<br><br>B – Best Possible Method [US Only]<br><br>Chase Paymentech utilizes the method that best fits the situation. If the RDFI is not an ACH participant, a facsimile draft will be created. This should be the default value for this field.<br><br>A – ACH [US or Canadian]<br><br>Deposit the transaction by ACH only. If the RDFI is not an ACH participant, the transaction is rejected.<br><br>*This is the equivalent to the <BankPmtDelv> element used on transactional requests.* | C | 1 | A |
| **SwitchSoloStartDate** | Profile | **Customer Switch / Solo cards activation date:**<br><br>Mandatory if Customer Profile Action Type = Create and the Customer Payment Type = Switch / Solo<br><br>Optional if Customer Profile Action Type = Update and the Customer Payment Type = Switch / Solo<br><br>- Format: MMYY<br><br>*This is the equivalent to the <DebitCardStartDate> element used on transactional requests.* | C | 4 | N |

| XML Label | XML Parent Tag | Description | Req. M/C/O | Max. Char. | Field Type A/N |
|---|---|---|---|---|---|
| **SwitchSoloIssueNum** | Profile | **Customer Switch / Solo cards Issue Number:**<br>Mandatory if Customer Profile Action Type = Create and the Customer Payment Type = Switch / Solo<br>Optional if Customer Profile Action Type = Update and the Customer Payment Type = Switch / Solo<br>Switch / Solo incremental counter for lost or replacement cards.<br>*This is the equivalent to the <DebitCardIssueNum> element used on transactional requests.* | C | 2 | N |

# Profile Response Elements

| XML Label | XML Parent Tag | Description | Req. M/C/O | Max. Char. | Field Type A/N |
|---|---|---|---|---|---|
| **Request** | N/A | **Required XML Parent Tag** | M | N/A | N/A |
| **profileResp** | Request | **XML tag that defines the transaction as an Batch request** | M | N/A | N/A |
| **CustomerBin** | profileResp | **Transaction Routing Definition:**<br>Value from the Request Returned | M | 6 | N |
| **CustomerMerchantID** | profileResp | **Gateway merchant account number from Request** | M | 15 | N |
| **CustomerName** | profileResp | **Customer Billing Name** | M | 30 | A |
| **CustomerRefNum** | profileResp | **Customer Reference Number** | M | 22 | A |
| **CustomerProfileAction** | profileResp | **Defines the Customer Profile action requested** | M | 6 | A |
| **CustomerProfileMessage** | profileResp | **Verbose Text Description associated with ProfileProcStatus** | M | Var | A |
| **CustomerAddress1** | profileResp | **Cardholder Billing Address line 1** | M | 30 | A |
| **CustomerAddress2** | profileResp | **Cardholder Billing Address line 2** | M | 30 | A |
| **CustomerCity** | profileResp | **Cardholder Billing City** | M | 20 | A |
| **CustomerState** | profileResp | **Cardholder Billing State** | M | 2 | A |
| **CustomerZIP** | profileResp | **Cardholder Billing Address Zip Code:** | M | 10 | A |
| **CustomerEmail** | profileResp | **Cardholder Email** | M | 50 | A |
| **CustomerPhone** | profileResp | **Cardholder Telephone Number:**<br>Format = AAAEEENNNNXXXX, where<br>AAA – Area Code<br>EEE – Exchange<br>NNNN – Number<br>XXXX – EXTENSION | M | 14 | A |

| XML Label | XML Parent Tag | Description | Req. M/C/O | Max. Char. | Field Type A/N |
|---|---|---|---|---|---|
| **CustomerProfileOrderOverrideInd** | profileResp | **Defines if any Order Data can be pre-populated from the Customer Reference Number [CustomerRefNum]:**<br>**NO** No mapping to order data<br>**OI** Use <*CustomerRefNum*> for <*OrderID*> and <*ECOrderNum*> or <*MailOrderNum*><br>**OD** Use <*CustomerReferNum*> for <*Comments*><br>**OA** Use <*CustomerRefNum*> for <*OrderID*> and <*ECOrderNum*> [or <*MailOrderNum*>] and <*Comments*> | M | 2 | A |
| **OrderDefaultDescription** | profileResp | **Order Description** | M | 64 | A |
| **OrderDefaultAmount** | profileResp | **Defaulted Transaction Amount**<br>Implied decimal | M | 12 | N |
| **CustomerAccountType** | profileResp | **Defines the Customers Payment Type**<br>C – Credit Card<br>S – Switch / Solo<br>E – Electronic Check | M | 1 | A |
| **CCAccountNum** | profileResp | **Customer Credit Card Number** | M | 19 | N |
| **CCExpireDate** | profileResp | **Customer Credit Card Expiration Date** | M | 4 | N |
| **ECPAccountDDA** | profileResp | **ECP [DDA] Account Number** | M | 17 | A |
| **ECPAccountType** | profileResp | **Defines the ECP deposit account type**<br>C – Consumer Checking [US or Canadian]<br>S – Consumer Savings [US Only]<br>X – Commercial Checking [US Only] | M | 1 | A |
| **ECPAccountRT** | profileResp | **ECP Bank routing and transit number for the customer** | M | 9 | N |

| XML Label | XML Parent Tag | Description | Req. M/C/O | Max. Char. | Field Type A/N |
|---|---|---|---|---|---|
| **ECPBankPmtDlv** | profileResp | **Defines the ECP payment delivery method**<br>B – Best Possible Method [US Only]<br>Chase Paymentech utilizes the method that best fits the situation. If the RDFI is not an ACH participant, a facsimile draft will be created. This should be the default value for this field.<br>A – ACH [US or Canadian]<br>**Deposit the transaction by ACH only. If the RDFI is not an ACH participant, the transaction is rejected.** | M | 1 | A |
| **SwitchSoloStartDate** | profileResp | **Customer Switch / Solo cards activation date**<br>Format: MMYY | M | 4 | N |
| **SwitchSoloIssueNum** | profileResp | **Customer Switch / Solo cards Issue Number** | M | 2 | N |

# FlexCache Request Elements

| XML Name | XML Parent Element | Description | Req. M/C/O | Max. Char. | Field Type A/N |
|---|---|---|---|---|---|
| **Request** | N/A | **Required XML Parent Tag** | M | N/A | N/A |
| **flexCache** | Request | **XML tag that defines the transaction as a New Order request** | M | N/A | N/A |
| **BIN** | flexCache | **Transaction Routing Definition:**<br>Assigned by Chase Paymentech<br>000001 – Salem<br>000002 – PNS | M | 6 | N |
| **MerchantID** | flexCache | **Gateway merchant account number assigned by Chase Paymentech:**<br>This account number will match that of your host platform<br>BIN 000001 – 6 digit Salem Division Number<br>BIN 000002 – 12 digit PNS Merchant ID | M | 12 | N |
| **TerminalID** | flexCache | **Merchant Terminal ID assigned by Chase Paymentech:**<br>All Salem Terminal IDs at present must be '001'. PNS Terminal ID's can be from '001' – '999'. Most are '001'. | M | 3 | N |
| **AccountNum** | flexCache | **Card Number identifying the customer.**<br>Should be null for electronic check processing | M | 19 | N |
| **OrderID** | flexCache | **Merchant Defined Order Number:**<br>Field defined and supplied by the auth originator, and echoed back in response.<br>The first 8 characters should be unique for each transaction.<br>Valid Characters:<br>abcdefghijklmnopqrstuvwxyzABCDEFGHIJKLMNOPQRSTUVWXYZ0123456789-,$@ and a space character. However, the space character cannot be the leading character. | M | 22 | A |

| XML Name | XML Parent Element | Description | Req. M/C/O | Max. Char. | Field Type A/N |
|---|---|---|---|---|---|
| **Amount** | flexCache | **Transaction Amount**:<br>Keys:<br>- Implied Decimal including those currencies that are a zero exponent. For example, both $100.00 (an exponent of '2') and 100 Yen (an exponent of '0') should be sent as <Amount>10000</Amount>.<br>- See table for min/max amount for each currency type. | M | 12 | N |
| **CardSecVal** | flexCache | **Card Verification Data [CVD] / PIN**<br>While the CVD value can be submitted on any transaction type, the FlexCache Host will only validate the value on the following transaction types:<br>- Authorize<br>- Redemption<br>- Balance Inquiry | O | 4 | N |
| **Comments** | flexCache | **Free form comments:**<br>Merchant can fill in this field and the info will be stored with the transaction details. | O | 64 | A |
| **ShippingRef** | flexCache | **Shipping Tracking Reference Number.**<br>Merchant can fill in this field and the info will be stored with the transaction details. | O | 40 | A |
| **IndustryType** | flexCache | **Defines the Industry type of the transaction:**<br>MO – Mail Order transaction<br>RC – Recurring Payment<br>EC– eCommerce transaction | M | 2 | A |

| XML Name | XML Parent Element | Description | Req. M/C/O | Max. Char. | Field Type A/N |
|---|---|---|---|---|---|
| **FlexPartialRedemptionInd** | flexCache | **Trigger to allow an approval for a Redemption Completion FlexAction if the available balance is less than the requested amount.**<br>Y – Approve Redemption Completion<br>Do not submit this Element if the desire is to not allow an Approval on a Redemption Completion unless the full amount on the card is available.<br>Only supported for Salem Interfaces | O | 1 | A |
| **FlexAction** | flexCache | **Defines the Transaction [or Action] Type:**<br>The following are the values that can be submitted:<br>- Activate<br>- DeActivate<br>- ReActivate<br>- AddValue<br>- Auth<br>- Redemption<br>- RedemptionCompletion<br>- Refund<br>- BalanceInquiry<br>- Void | M | 30 | A |
| **StartAccountNum** | flexCache | **Defines the first Card Number in a Block Activation Sequence** | C | 19 | N |
| **ActivationCount** | flexCache | **Defines the number of Cards in addition to the first Card Number in the sequence**<br>- The maximum number of cards that can be activated at one time is 100<br>- As such, the maximum number allowed to be sent in this field is 99. | M | 2 | N |
| **TxRefNum** | flexCache | **Gateway transaction reference number:**<br>A unique value for each transaction, which is required to adjust any transaction in the gateway [such as Mark for Capture or void]. | M | 40 | A |

| XML Name | XML Parent Element | Description | Req. M/C/O | Max. Char. | Field Type A/N |
|---|---|---|---|---|---|
| **FlexEmployeeNumber** | flexCache | **Employee Number**<br>Optionally available field to pass an Employee Number on the transaction. This will appear in FlexCache generated [not Orbital Gateway] reports.<br>Only supported for PNS Interfaces | O | 15 | A |
| **PriorAuthID** | flexCache | **Prior Authorization Code:**<br>If a prior authorization code is available, It should be sent in this tag. This reduces the risk of chargebacks.<br>- This element should not be included on an ECP transaction<br>- This should only be sent if the transType = F or FC<br>Only supported for PNS Interfaces | O | 6 | A |

# FlexCache Response Elements

| XML Name | XML Parent Element | Description | Req. M/C/O | Max. Char. | Field Type A/N |
|---|---|---|---|---|---|
| **Request** | N/A | **Required XML Parent Tag** | M | N/A | N/A |
| **flexCacheResp** | Request | **XML tag that defines the transaction as a New Order request** | M | N/A | N/A |
| **MerchantID** | flexCacheResp | **Gateway merchant account number assigned by Chase Paymentech:**<br>This account number will match that of your host platform<br>BIN 000001 – 6 digit Salem Division Number<br>BIN 000002 – 12 digit PNS Merchant ID | M | 12 | N |
| **TerminalID** | flexCacheResp | **Merchant Terminal ID assigned by Chase Paymentech:**<br>All Salem Terminal IDs at present must be '001'. PNS Terminal ID's can be from '001' – '999'. Most are '001'. | M | 3 | N |
| **OrderID** | flexCacheResp | **Merchant Defined Order Number:**<br>Field defined and supplied by the auth originator, and echoed back in response. | M | 22 | A |
| **AccountNum** | flexCacheResp | **Card Number identifying the customer.**<br>Should be null for electronic check processing | C | 19 | N |
| **StartAccountNum** | flexCacheResp | **Defines the first Card Number in a Block Activation Sequence** | C | 19 | N |
| **BatchFailedAcctNum** | flexCacheResp | **Card Number in a Block Activation Sequence that caused a Block Activation Failure**<br>Conditionally returned on a Block Activation failure | C | 19 | N |
| **flexRequestedAmount** | flexCacheResp | **Transaction amount submitted in the request**:<br>Implied Decimal.<br>Conditionally returned | C | 12 | N |

| XML Name | XML Parent Element | Description | Req. M/C/O | Max. Char. | Field Type A/N |
|---|---|---|---|---|---|
| **flexRedeemedAmt** | flexCacheResp | **Actual amount Redeemed on a Redemption Completion where the flexPartialRedemptionInd = Yes**:<br>Implied Decimal.<br>Conditionally returned Regardless of whether the amount redeemed is equal to or less than the requested amount, it will be identified in this tag. | C | 12 | N |
| **flexHostTrace** | flexCacheResp | **Gateway transaction reference number:**<br>A unique value for each transaction, which is required to adjust any transaction in the gateway [such as Mark for Capture or void/Reversal]. | C | 40 | N |
| **flexAction** | flexCacheResp | **Returns the Transaction [or Action] Type performed from the request:**<br>The FlexAction returned in the response is the same value submitted in the request. | M | 30 | A |
| **flexAcctBalance** | flexCacheResp | **Current Balance of the FlexCache Card**<br>The Balance after the result of the request transaction. This information will returned in all FlexCache response messages. | M | 12 | N |
| **flexAcctPriorBalance** | flexCacheResp | **Prior Balance of the FlexCache Card**<br>Balance prior to the result of the request transaction. This information will returned in all FlexCache response messages. | M | 12 | N |
| **flexAcctExpireDate** | flexCacheResp | **The FlexCache Card Expiration Date**<br>The Expiration Date of the FlexCache will be returned, if one exists.<br>This information will returned in all response messages.<br>Response Format = MMMMYY | M | 6 | N |
| **CardBrand** | flexCacheResp | **Mnemonic representing the of the request card type:**<br>FC – FlexCache | M | 2 | A |

| XML Name | XML Parent Element | Description | Req. M/C/O | Max. Char. | Field Type A/N |
|---|---|---|---|---|---|
| **TxRefNum** | flexCacheResp | **Gateway transaction reference number:**<br><br>A unique value for each transaction, which is required to adjust any transaction in the gateway. | M | 40 | A |
| **TxRefIdx** | flexCacheResp | **Gateway transaction index:**<br><br>Used to identify the unique components of transactions adjusted more than one time. Required on for void transactions. | M | 4 | N |
| **ProcStatus** | flexCacheResp | **Process Status:**<br><br>The first data set that should be checked to determine the result of a request. It is the only element that is returned in all response scenarios. It identifies whether transactions have successfully passed all of the Gateway edit checks.<br><br>0 – Success<br><br>All other values constitute an error condition and will be returned in a SOAPFault. See appendix for definition of those error values. | M | 6 | A |
| **ApprovalStatus** | flexCacheResp | **Approval Status:**<br><br>Conditional on:<br><br>- Process Status returning a '0' or successful response.<br>- Only Returned if performing a MFC on a FlexCache Card Type<br><br>If so, approval status identifies the result of the authorization request to the host system.<br><br>0 – Decline<br>1 – Approved<br>2 – Message/System Error | M | 1 | N |
| **AuthCode** | flexCacheResp | **Issuer approval Code:**<br><br>Unique transactional level code issuer uses to show each request was approved | M | 6 | A |

| XML Name | XML Parent Element | Description | Req. M/C/O | Max. Char. | Field Type A/N |
|---|---|---|---|---|---|
| **RespCode** | flexCacheResp | **Response Code:**<br>Normalized authorization response code issued by the host system [Salem / PNS], which identifies an approval ('00') or the reason for a decline or error.<br>Conditionally returned when procStatus = 0 | M | 2 | A |
| **CVV2RespCode** | flexCacheResp | **Card verification value request response:**<br>- See appendix for values<br>- Conditional on card verification request being sent | M | 1 | A |
| **RespTime** | flexCacheResp | **Time the transaction was processed by gateway:**<br>Format – 'hh24miss' | M | 6 | N |

# Quick Response Elements

Note: For any request when the message cannot be delivered to the Host Authorization or Capture Systems, A Quick Response message may be issued. The relevant fields are:

| XML Name | XML Parent Element | Description | Req. M/C/O | Max. Char. | Field Type A/N |
|---|---|---|---|---|---|
| **Response** | N/A | **Required XML Parent Tag** | M | N/A | N/A |
| **QuickResp** | Response | **XML tag that defines the transaction as a New Order response** | M | N/A | N/A |
| **MerchantID** | QuickResp | **Gateway merchant account number assigned by Chase Paymentech Solutions:**<br>Echoes the account number passed in the request | M | 12 | N |
| **TerminalID** | QuickResp | **Merchant Terminal ID assigned by Chase Paymentech Solutions:**<br>Echoes the Terminal ID passed in the request | M | 3 | N |
| **OrderID** | QuickResp | **Merchant Defined Order Number:**<br>Echoes the Order Number passed in the request | C | 22 | A |
| **AccountNum** | QuickResp | **Card Number identifying the customer.**<br>Echoes the Account Number passed in the request | C | 19 | N |
| **TxRefNum** | QuickResp | **Gateway transaction reference number:**<br>Conditionally returned dependant on the error | C | 40 | A |
| **TxRefIdx** | QuickResp | **Gateway transaction index:** | C | 4 | A |
| **ProcStatus** | QuickResp | **Process Status:**<br>The first element that should be checked to determine the result of a request. It is the only element that is returned in all response scenarios. It identifies whether transactions have successfully passed all of the Gateway edit checks.<br>See appendix for definition of those error values. | M | 6 | A |
| **StatusMsg** | QuickResp | **Text message associated with ProcStatus value.** | C | Var | A |

| XML Name | XML Parent Element | Description | Req. M/C/O | Max. Char. | Field Type A/N |
|---|---|---|---|---|---|
| **RespTime** | QuickResp | **Time the transaction was processed by gateway:**<br>Format – 'hh24miss' | M | 6 | N |

# *Appendix A*

## RespCode Tag Values

Response codes received that are not in this table should be treated as a general error not approved.

**Key:**

| Action | Description |
|---|---|
| Call | Call your Chase Paymentech Customer Service for assistance |
| Cust. | Try to resolve with customer or obtain alternate payment method |
| Fix | There is an invalid value being sent. Fix and resend |
| Resend | Send this transaction back at any time |
| Voice | Perform a voice authorization per instructions provided by Chase Paymentech |
| Wait | Wait 2-3 days before resending or try to resolve with the customer |

**Response Codes:**

| Code | Definition | Status | Action |
|---|---|---|---|
| 00 | Approved | Approved | None |
| 01 | Call/Refer to Card Issuer | Decline | Voice |
| 02 | Refer to Card issuer's special conditions | Decline | Voice |
| 03 | Invalid Merchant Number | Error | Fix |
| 04 | Pickup | Decline | Cust. |
| 05 | Do Not Honor | Decline | Cust. |
| 06 | Other Error | Decline | Cust. |
| 07 | Stop Deposit Order | Decline | Cust. |
| 08 | Approved authorization, honor with identification | Approved | None |
| 09 | Revocation of Auth | Decline | Cust. |
| 10 | Default Call | Decline | Voice |
| 11 | Approved authorization, VIP Approval | Approved | None |
| 12 | Invalid Transaction Type | Decline | Cust. |
| 13 | Bad Amount | Decline | Fix |
| 14 | Invalid Credit Card Number | Decline | Fix |

| Code | Definition | Status | Action |
|---|---|---|---|
| 15 | Default Call Low Fraud | Decline | Voice |
| 16 | Default Call Medium Fraud | Decline | Voice |
| 17 | Default Call High fraud | Decline | Voice |
| 18 | Default Call Unavailable Fraud | Decline | Voice |
| 19 | Re-enter Transaction | Error | Resend |
| 20 | Floor Low Fraud | Decline | Cust. |
| 21 | Floor Medium Fraud | Decline | Cust. |
| 22 | Floor High fraud | Decline | Cust. |
| 23 | Floor Unavailable Fraud | Decline | Cust. |
| 24 | Validated | Approved | None |
| 26 | Pre-noted | Approved | None |
| 27 | No reason to decline | Decline | Cust. |
| 28 | Received and stored | Approved | None |
| 29 | Provided Auth | Approved | None |
| 30 | Invalid value in message | Error | Fix |
| 31 | Request received | Approved | None |
| 32 | BIN Alert | Approved | None |
| 33 | Card is expired | Decline | Cust. |
| 34 | Approved for partial | Approved | None |
| 35 | Zero Amount | Error | Fix |
| 36 | Bad Total Auth amount | Error | Fix |
| 37 | Invalid Secure Payment Data | Error | Fix |
| 38 | Merchant not MC SecureCode Enabled | Decline | Call |
| 40 | Requested Function not supported | Error | Call or Fix |
| 41 | Lost / Stolen | Decline | Cust. |
| 42 | Account Not Active | Decline | Cust. |
| 43 | Lost / Stolen Card | Decline | Cust. |
| 45 | Duplicate Transaction | Decline | Cust. |
| 46 | Blanks not passed in Reserved Field | Decline | Fix |
| 50 | Positive ID | Decline | Cust. |
| 52 | Processor Decline | Decline | Cust. |
| 56 | Restraint | Decline | Cust. |
| 58 | Transaction not permitted to terminal | Error | Call |
| 59 | Soft AVS | Decline | Cust. |
| 60 | Do Not Honor Low Fraud | Decline | Cust. |

| Code | Definition | Status | Action |
|---|---|---|---|
| 61 | Do Not Honor Medium Fraud | Decline | Cust. |
| 62 | Do Not Honor High fraud | Decline | Cust. |
| 63 | Do Not Honor Unavailable Fraud | Decline | Cust. |
| 64 | CVV2/CVC2 Failure | Decline | Cust. |
| 65 | Invalid Amex CID | Decline | Cust. |
| 66 | Other Error | Error | Fix |
| 68 | Invalid CC Number | Error | Fix |
| 69 | Does not match MOP | Error | Fix |
| 71 | No Account | Decline | Fix |
| 72 | Invalid Institution Code | Decline | Fix |
| 73 | Method of Payment is Invalid for Merchant | Error | Fix |
| 74 | Invalid Expiration Date | Decline | Cust. |
| 75 | Bad Amount | Error | Fix |
| 77 | Invalid Amount | Decline | Fix |
| 78 | Missing Companion Data | Error | Fix |
| 79 | Invalid Merchant | Error | Fix |
| 80 | Invalid MOP for division | Error | Fix |
| 81 | Call Low Fraud | Decline | Voice |
| 82 | Call Medium Fraud | Decline | Voice |
| 83 | Call High Fraud | Decline | Voice |
| 84 | Call Unavailable Fraud | Decline | Voice |
| 85 | Duplicated Order # | Error | Fix |
| 86 | Auth Recycle Host down | Error | Wait |
| 87 | Invalid Currency | Error | Fix |
| 88 | Invalid Purch. Level III | Error | Fix |
| 89 | Credit Floor | Decline | Cust. |
| 91 | Approved Low Fraud | Approved | None |
| 92 | Approved Medium Fraud | Approved | None |
| 93 | Approved High Fraud | Approved | None |
| 94 | Approved Fraud Service Unavailable | Approved | None |
| 95 | Invalid Data Type | Error | Fix |
| 96 | Invalid Record Sequence | Error | Fix |
| 97 | Percents Not Total 100 | Error | Fix |
| 98 | Issuer Unavailable | Decline | Resend |
| 99 | No Answer / Unable to send | Error | Resend |

| Code | Definition | Status | Action |
|---|---|---|---|
| A1 | Payments Not Total Order | Error | Fix |
| A2 | Bad Order Number | Error | Fix |
| A3 | FPO Locked | Error | Wait |
| A4 | FPO Not Allowed | Error | Call |
| A5 | Auth Amount Wrong | Error | Fix |
| A6 | Illegal Action | Error | Fix |
| A8 | Invalid Start Date | Error | Fix |
| A9 | Invalid Issue Number | Error | Fix |
| B1 | Invalid Transaction Type | Error | Fix |
| B2 | Account Previously Activated | Decline | Cust |
| B3 | Unable to void transaction | Error | Fix |
| B5 | Not on file | Decline | Fix |
| B7 | Fraud | Decline | Cust. |
| B8 | Bad Debt | Decline | Cust. |
| B9 | On Negative File | Decline | Cust. |
| BA | Under 18 Years Old | Decline | Cust. |
| BB | Possible Compromise | Decline | Cust. |
| BC | Bill To Not Equal To Ship To | Decline | Cust. |
| BD | Invalid Pre-approval Number | Decline | Cust. |
| BE | Invalid Email Address | Decline | Cust. |
| BF | PA ITA Number Inactive | Decline | Cust. |
| BG | Blocked Account | Decline | Cust. |
| BH | Address Verification Failed | Decline | Cust. |
| Bl | Not on Credit Bureau | Decline | Cust. |
| BJ | Previously Declined | Decline | Cust. |
| BK | Closed Account, New Account Closed | Decline | Cust. |
| BL | Re-Authorization | Decline | Cust. |
| BM | Re-Authorization – No Match | Decline | Cust. |
| BN | Re-Authorization – Timeframes Exceeded | Decline | Cust. |
| BO | Stand In Rules | Decline | Cust. |
| BP | 'Customer Service Phone Number required on Transaction Types 1 (MO/TO) and 2 (Recurring). MC Only | Error | Fix |
| BQ | 'Issuer has flagged account as suspected fraud. (Discover Only) | Decline | Cust. |
| BR | Invalid MCC Sent | Error | Fix |

| Code | Definition | Status | Action |
|---|---|---|---|
| BS | New Card Issued | Decline | Cust. |
| C1 | Invalid Issuer | Decline | Cust. |
| C2 | Invalid Response Code | Decline | Fix |
| C3 | Excessive Pin Try | Decline | Cust. |
| C4 | Over Limit | Decline | Cust. |
| C5 | Over Freq Limit | Decline | Cust. |
| C6 | Over Sav limit | Decline | Cust. |
| C7 | Over Sav Freq | Decline | Cust. |
| C8 | Over Credit limit | Decline | Cust. |
| C9 | Over Credit Freq | Decline | Cust. |
| D1 | Invalid For Credit | Decline | Fix |
| D2 | Invalid For Debit | Decline | Fix |
| D3 | Rev Exceed Withdrawal | Decline | Cust. |
| D4 | One Purchasing Limit | Decline | Cust. |
| D5 | On Negative File | Decline | Cust. |
| D6 | Changed Field | Decline | Fix |
| D7 | Insufficient Funds | Decline | Cust. |
| D8 | Encrypted Data Bad | Decline | Fix |
| D9 | Altered Data | Decline | Fix |
| E3 | Invalid Prefix | Decline | Fix |
| E4 | Invalid Institution | Decline | Fix |
| E5 | Invalid Cardholder | Decline | Fix |
| E6 | BIN Block | Decline | Fix |
| E7 | Stored | Approved | None |
| E8 | Invalid Transit Routing Number | Error | Fix |
| E9 | Unknown Transit Routing Number | Error | Fix |
| F1 | Missing Name | Error | Fix |
| F2 | Invalid Account Type | Error | Fix |
| F3 | Account Closed | Error | Cust. |
| F4 | No Account / Unable To Locate | Error | Fix |
| F5 | Account holder Deceased | Error | Cust. |
| F6 | Beneficiary Deceased | Error | Cust. |
| F7 | Account Frozen | Error | Cust. |
| F8 | Customer Opt Out | Error | Cust. |
| F9 | ACH Non-Participant | Error | Cust. |

| Code | Definition | Status | Action |
|---|---|---|---|
| G1 | No Pre-note | Error | Fix |
| G2 | No Address | Error | Fix |
| G3 | Invalid Account Number | Error | Fix |
| G4 | Authorization Revoked by Consumer | Error | Cust. |
| G5 | Customer Advises Not Authorized | Error | Cust. |
| G6 | Invalid CECP Action Code | Error | Fix |
| G7 | Invalid Account Format | Error | Fix |
| G8 | Bad Account Number Data | Error | Fix |
| G9 | No Capture | Decline | N/A |
| H1 | No Credit Function | Decline | N/A |
| H2 | No Debit Function | Decline | N/A |
| H3 | Rev Exceed Withdrawal | Decline | Cust. |
| H4 | Changed Field | Decline | N/A |
| H5 | Terminal Not Owned | Decline | N/A |
| H6 | Invalid Time | Decline | Fix |
| H7 | Invalid Date | Decline | Fix |
| H8 | Invalid Terminal Number | Decline | Fix |
| H9 | Invalid PIN | Decline | Cust. |
| I1 | Block activation failed – Card Range not set up for MOD 10 | Error | Fix |
| I2 | Block activation failed – email or fulfillment flags were set to Y | Error | Fix |
| I3 | Declined – Issuance doesn't meet minimum amount | Declined | Cust |
| I4 | Declined – no original auth found | Decline | Cust |
| I5 | Declined – outstanding auth, funds on hold | Decline | Cust |
| I6 | Activation amount incorrect | Decline | Fix |
| I7 | Block activation failed- account not correct or block size not correct | Decline | Fix |
| I8 | Mag stripe CVD value failed | Decline | Fix |
| I9 | Max Redemption limit met | Decline | Fix |
| J1 | No Manual Key | Decline | Fix |
| J2 | Not Signed In | Decline | Fix |
| J3 | Excessive Pin Try | Decline | Cust. |
| J4 | No DDA | Decline | Fix |
| J5 | No SAV | Decline | Fix |

| Code | Definition | Status | Action |
|---|---|---|---|
| J6 | Excess DDA | Decline | Cust. |
| J7 | Excess DDA FREQ | Decline | Cust. |
| J8 | Excess SAV | Decline | Cust. |
| J9 | Excess SAV FREQ | Decline | Cust. |
| K1 | Excess Card | Decline | Cust. |
| K2 | Excess Card Freq | Decline | Cust. |
| K3 | Reserved Future | Decline | N/A |
| K4 | Reserved Closing | Decline | N/A |
| K5 | Dormant | Decline | Cust. |
| K6 | NSF | Decline | Cust. |
| K7 | Future RD Six | Decline | N/A |
| K8 | Future RD Seven | Decline | N/A |
| K9 | Transaction Code Conflict | Decline | Fix |
| L1 | In Progress | Decline | Wait |
| L2 | Process Unavailable | Error | Resend |
| L3 | Invalid Expiration | Error | Fix |
| L4 | Invalid Effective | Error | Fix |
| L5 | Invalid Issuer | Decline | Fix |
| L6 | Tran not allowed for cardholder | Decline | Cust. |
| L7 | Unable to Determine Network Routing | Error | Call |
| L8 | System Error | Error | Call |
| L9 | Database Error | Error | Call |
| M1 | Merchant Override Decline | Decline | Cust. |
| PA | Partial Approval | Approved | N/A |
| PB | Revocation of all Authorization | Decline | Cust. |

# AVSRespCode Tag Values

| Code | AVS Message |
|---|---|
| 1 | No address supplied |
| 2 | Bill-to address did not pass Auth Host edit checks |
| 3 | AVS not performed |
| 4 or R | Issuer does not participate in AVS |
| 5 | Edit-error - AVS data is invalid |

| | |
|---|---|
| 6 | System unavailable or time-out |
| 7 | Address information unavailable |
| 8 | Transaction Ineligible for AVS |
| 9 | Zip Match / Zip4 Match / Locale match |
| A | Zip Match / Zip 4 Match / Locale no match |
| B | Zip Match / Zip 4 no Match / Locale match |
| C | Zip Match / Zip 4 no Match / Locale no match |
| D | Zip No Match / Zip 4 Match / Locale match |
| E | Zip No Match / Zip 4 Match / Locale no match |
| F | Zip No Match / Zip 4 No Match / Locale match |
| G | No match at all |
| H | Zip Match / Locale match |
| J | Issuer does not participate in Global AVS |
| JA | International street address and postal match |
| JB | International street address match. Postal code not verified. |
| JC | International street address and postal code not verified. |
| JD | International postal code match. Street address not verified. |
| M1 | Cardholder name matches |
| M2 | Cardholder name, billing address, and postal code matches |
| M3 | Cardholder name and billing code matches |
| M4 | Cardholder name and billing address match |
| M5 | Cardholder name incorrect, billing address and postal code match |
| M6 | Cardholder name incorrect, billing address matches |
| M7 | Cardholder name incorrect, billing address matches |
| M8 | Cardholder name, billing address and postal code are all incorrect |
| N3 | Address matches, ZIP not verified |
| N4 | Address and ZIP code match (International only) |
| N5 | Address not verified (International only) |
| N6 | Address and ZIP code match (International only) |
| N7 | ZIP matches, address not verified |
| N8 | Address and ZIP code match (International only) |
| UK | Unknown |
| X | Zip Match / Zip 4 Match / Address Match |
| Y | Not Performed |
| Z | Zip Match / Locale no match |

| | |
|---|---|
| 'Blank' | Not applicable (non-Visa) |

# CVV2RespCode Tag Values

| Code | CVV2 (CVC2/ Discover CID) Description |
|---|---|
| M | CVV Match |
| N | CVV No match |
| P | Not processed |
| S | Should have been present |
| U | Unsupported by issuer |
| I | Invalid |
| Y | Invalid |
| 'Blank' | Not applicable (non-Visa) |

# CAVVRespCode Tag Values

| Code | CAVV Description |
|---|---|
| 'Blank' | CAVV Not Present |
| 0 | CAVV Not Validated due to erroneous data submitted |
| 1 | CAVV Failed Validation – Authentication Transaction |
| 2 | CAVV Passed Validation – Authentication Transaction |
| 3 | CAVV Attempt a 3-D Secure authentication value of 7 from the Issuer ACS indicates authentication was attempted. (Determined that the issuer ACS generated this value from the use of Visa CAVV key[s]). |
| 4 | CAVV Failed Validation – Attempt a 3-D Secure authentication value of 7 from Visa's ACS indicates that an authentication attempt was performed. (Determined that Visa generated this value from the use of CAVV key[s]). |
| 5 | Reserved for Future Use – NOT USED. |
| 6 | CAVV Not Validated – Issuer not participating in CAVV validation. |
| 7 | CAVV Failed Validation - Attempt (CAVV generated with Visa Key) |
| 8 | CAVV Passed Validation – Attempt (CAVV generated with Visa Key) |

| | |
|---|---|
| 9 | CAVV Failed Validation - Attempt (CAVV generated with Visa Key – Issuer ACS unavailable) |
| A | CAVV Passed Validation – Attempt (CAVV generated with Visa Key – Issuer ACS unavailable) |
| B | CAVV Passed Validation – Information only, no liability shift (CAVV with ECI = 7) |
| C | CAVV Not Validated – Attempt Issuer did not return a CAVV results code in the Authorization response. |
| D | CAVV Not Validated – Authentication – Issuer did not return a CAVV results code in the authorization response. |
| I | Invalid Security Data |
| U | Issuer does not participate or 3-D Secure data not utilized. |

# HTTP Responses

A few listing excluding specific Orbital Gateway HTTP responses and the descriptions can be found at http://www.w3.org/Protocols/rfc2616/rfc2616.txt. The Gateway specific and common HTTP responses are:

| Code | Definition | Status |
|---|---|---|
| 200 | Approved | An HTTP Session was established with the Orbital Gateway. Error conditions can still be returned. |
| 400 | Invalid Request | The server, due to malformed syntax, could not understand the request. |
| 403 | Forbidden: SSL Connection Required | A Clear Text [or unencrypted] request was made to the Orbital Gateway. All transactions must be SSL Encrypted to interface to Orbital. |
| 408 | Request Timed Out | The Response could not be processed within the maximum time allowed. |
| 412 | IP Security Failure | A non-registered IP Address attempted to connect to the Orbital Gateway. The HTTP connection was refused as a result. |
| 500 | Internal Server Error | The server encountered an unexpected condition, which prevented it from fulfilling the request. |
| 502 | Connection Error | The server, while acting as a gateway or proxy, received an invalid response from the upstream server it accessed in attempting to fulfill the request. |

# ProcStatus Tag Error Values

| Code | Description | Action |
|---|---|---|
| 1 | PWS_UNKNOWN_ERROR | Resend |

| Code | Description | Action |
|---|---|---|
| 2 | PWS_NETWORK_ERROR | Resend |
| 3 | PWS_DB_ERROR<br>Unknown Database Issues | Resend |
| 40 | Cannot Get to Authorizer Service | Resend |
| 54 | Industry type is currently not supported for merchant and bin | Fix |
| 205 | PWS_DB_EXCEPTION_ERROR | Resend |
| 208 | PWS_ERROR_FAILED_TO_CONNECT | Resend |
| 301 | PWS_NW_OPEN_ERROR | Resend |
| 303 | PWS_NW_READ_ERROR | Resend |
| 328 | PWS_ERROR_BAD_REVERSAL_AMOUNT<br>(An invalid amount submitted on a Partial Void Request) | Fix |
| 329 | PWS_ERROR_BAD_REQUEST_AMOUNT | Fix |
| 330 | PWS_ERROR_ALREADY_CAPTURED | Fix |
| 331 | PWS_ERROR_INVALID_ACTION | Fix |
| 333 | PWS_ERROR_MISSING_TRANSACTION_REFERENCE_INDEX | Fix |
| 335 | PWS_ERROR_SPLIT_AUTH_NOT_ALLOWED_ALREADY_MARKED | Fix |
| 348 | PWS_DID_NOT_ALLOW_A_CAPTURE_REQUEST_BECAUSE_THE_ORIGINAL_AUTH_WAS_NOT_SUCCESSFUL<br>(Cannot Void a Transaction in which the Mark for Capture Failed) | Fix |
| 350 | The amount requested cannot be zero | Fix |
| 351 | This industry type does not allow a capture greater than the value of the auth | Fix |
| 355 | There is nothing to capture [This error is returned when a Capture attempt is made on prior authorization, but there is no amount left to capture]. | Fix |
| 400 | PWS_MANDATORY_FIELDS_ERROR | Fix |
| 410 | FE_NETWORK_ERROR (cannot connect to eHost) | Resend |
| 411 | FE_INTERRUPTED_SESSION (i/o problem while connecting to eHost) | Resend |
| 516 | The Merchant ID / Acquiring BIN ID is invalid or missing. Message rejected | Fix |
| 518 | This merchant is not active until … [This error is returned when a Merchant Account has been setup, but with an Activation date in the future of the present date]. | Call Customer Service |
| 519 | This merchant is inactive | Call Customer Service |
| 521 | eHost has received a badly formatted message [This error is returned when required fields are missing] | Fix |
| 523 | An invalid TID was received {Terminal ID] | Fix |
| 801 | PWS_ERR_VALIDATION_AMOUNT | Fix |
| 803 | PWS_ERR_VALIDATION_AVSADDRESS | Fix |
| 804 | PWS_ERR_VALIDATION_AVSZIPCODE | Fix |

| Code | Description | Action |
|---|---|---|
| 806 | PWS_ERR_VALIDATION_BIN | Fix |
| 811 | PWS_ERR_VALIDATION_CUSTOMERADDR | Fix |
| 812 | PWS_ERR_VALIDATION_CUSTOMEREMAIL | Fix |
| 814 | PWS_ERR_VALIDATION_CUSTOMERNAME | Fix |
| 817 | PWS_ERR_VALIDATION_CUSTOMERPHONE | Fix |
| 818 | PWS_ERR_VALIDATION_CVV2 | Fix |
| 822 | PWS_ERR_VALIDATION_ISSUENUM | Fix |
| 823 | PWS_ERR_VALIDATION_LANGUAGE | Fix |
| 825 | PWS_ERR_VALIDATION_MERCHANTID | Fix |
| 826 | PWS_ERR_VALIDATION_ORDERDESCRIPTION | Fix |
| 827 | PWS_ERR_VALIDATION_ORDERID | Fix |
| 831 | PWS_ERR_VALIDATION_TAXAMT | Fix |
| 832 | PWS_ERR_VALIDATION_TAXINCLUDED | Fix |
| 833 | PWS_ERR_VALIDATION_TERMINALID | Fix |
| 834 | PWS_ERR_VALIDATION_TRANSDATE | Fix |
| 835 | PWS_ERR_VALIDATION_TRANSTIME | Fix |
| 836 | PWS_ERR_VALIDATION_ECOM | Fix |
| 838 | PWS_ERR_VALIDATION_ACNUMBER | Fix |
| 839 | PWS_ERR_VALIDATION_PAN_LUHN | Fix |
| 840 | PWS_ERR_VALIDATION_PAN_LENGTH | Fix |
| 841 | PWS_ERR_VALIDATION_PAN_RANGE | Fix |
| 842 | PWS_ERR_VALIDATION_EXP_DATE_FORMAT | Fix |
| 844 | PWS_ERR_VALIDATION_EXP_DATE_TOO_NEW | Fix |
| 845 | PWS_ERR_VALIDATION_START_DATE_FORMAT | Fix |
| 846 | PWS_ERR_VALIDATION_START_DATE_TOO_NEW | Fix |
| 847 | PWS_ERR_VALIDATION_PAN_FORMAT | Fix |
| 848 | PWS_ERR_VALIDATION_CURRENCY_FORMAT | Fix |
| 849 | PWS_ERR_VALIDATION_CURRENCY_UNSUPPORTED | Fix |
| 850 | PWS_ERR_VALIDATION_CURRENCY_BAD_EXPONENT | Fix |
| 851 | PWS_ERR_VALIDATION_MERCHANT_UNSUPPORTED | Fix |
| 852 | PWS_ERR_VALIDATION_BRAND_UNSUPPORTED | Fix |
| 853 | PWS_ERR_VALIDATION_BRAND_PAN_MISMATCH | Fix |
| 881 | The LIDM you supplied # does not match with any existing transaction (Cannot void or Mark a Transaction because the TxRefNum does match a transaction) | Fix |
| 882 | LOCKED_DOWN (Cannot mark or unmark transaction) | Fix |

| Code | Description | Action |
|---|---|---|
| 9720 | Soft Desc: Merchant not activated for soft descriptors | Fix |
| 9721 | Soft Desc: Merchant Name is required if soft descriptor data is sent | Fix |
| 9722 | Soft Desc: Merchant Name exceeds max length of [%s] for %s transactions | Fix |
| 9723 | Soft Desc: [%s] cannot contain leading spaces | Fix |
| 9724 | Soft Desc: [%s] exceeds max length of [%s] | Fix |
| 9725 | Soft Desc: Product Description cannot be present if Merchant Name is > %s | Fix |
| 9726 | Soft Desc: Product Description length cannot exceed [%s] if Merchant Name length is between %s and %s | Fix |
| 9727 | Soft Desc: Too many Merchant descriptors. Never send more than one of the following: City, phone, url OR email | Fix |
| 9728 | Soft Desc: [%s] is not allowed for ECP transactions | Fix |
| 9729 | Soft Desc: Invalid format for Merchant Phone. Must be nnn-nnn-aaaa or nnn-aaaaaaaa | Fix |
| 9737 | Gateway is Down | Resend |
| 9738 | Database Connection Problem: Cannot acquire Database Connection | Resend |
| 9743 | Pcard 3 data was sent in parent split, but is missing in current request | Fix |
| 9744 | If Alt Tax is sent Alt Tax ID is required | Fix |
| 9745 | Three reasons could result in this error:<br>- Pcard 3 data can only be sent with MC and VI cards.<br>- Pcard 3 data cannot be sent on this request type.<br>- Pcard 3 data can only be sent with US or Canadian currency. | Fix |
| 9746 | Line item count must be between 1 and 98 inclusive | Fix |
| 9747 | Line item detail number [%s] is missing | Fix |
| 9748 | Cannot send Pcard 3 data without sending Pcard 2 field | Fix |
| 9749 | Minimal Pcard 3 base data missing or invalid | Fix |
| 9750 | Minimal Pcard 3 line item data missing or invalid on index | Fix |
| 9751 | Line Item Count does not match the number of line items sent | Fix |
| 9752 | Invalid debit indicator for Bin 000002 in index. Must be 'D' or 'C' | Fix |
| 9753 | Invalid Gross / Net for Bin 000002 in index. Must be 'Y' or 'N' | Fix |
| 9754 | Amount hash error, negative total on line item data index | Fix |
| 9755 | Amount hash error on line item data index. Total = [%s] Hash = [%s] | Fix |
| 9756 | Detail totals do not match requested amount | Fix |
| 9757 | Invalid Country Code | Fix |
| 9758 | Invalid Unit of Measure in index | Fix |
| 9765 | The field is missing, invalid, or has exceeded the max length | Fix |

| Code | Description | Action |
|---|---|---|
| 9766 | The Bill Me Later Card Type [BL] is Not Allowed with this transaction. | Fix |
| 9767 | Bill Me Later Generic Error Code | Fix |
| 9768 | Invalid [Values. Must be one of the following values: XXX or empty | Fix |
| 9769 | BML: Mandatory Field [Customer Birth Date] is missing for [New (N)] Customer Type | Fix |
| 9793 | PINLess Debit: Invalid. The field is missing, invalid, or has exceeded the max length. | Fix |
| 9794 | PINLess Debit: The PINLess Debit Card Type [DP] is Not Allowed with [%s] Transactions. | Fix |
| 9795 | PINLess Debit: The PINLess Debit Card Type [DP] is Only Allowed with [%s] Transactions | Fix |
| 9796 | PINLess Debit: The PINLess Debit Card Type [DP] must be sent with Industry Type of [%s]. | Fix |
| 9797 | PINLess Debit: Card Number Not Eligible for PINLess Debit Processing | Fix |
| 10005 | Error communicating with the host | Fix |
| 10011 | Response timed out waiting for Authorization Host | Resend |
| 10096 | Invalid Card Number | Fix |
| 10204 | Invalid AVS ZIP Code | Fix |
| 10332 | Invalid Message Format. Transaction was flagged as an eCommerce Industry Type but No ECOrderNum was sent | Fix |
| 10333 | The ECOrderNum or MailOrderNum was all Zero's or All Spaces. These are not valid | Fix |
| 10334 | Invalid Card Number | Fix |
| 10336 | Transaction Amount to Large | Fix |
| 10337 | Transaction Amount to Small | Fix |
| 10349 | Host eFalcon check requested from PNS [BIN 000002] – This functionality is not supported on this platform | Fix |
| 11001 | Locked Down: Unable to Perform a Partial Void on Industry Type: [RE]. | Fix |
| All other 10000 – 11000 | GATEWAY SYSTEM ERROR CONDITIONS<br>This encompasses various processing errors. | Resend |
| | **Profile Errors** | |
| 9550 | Invalid Customer Reference Number From Order Indicator | Fix |
| 9551 | Invalid Customer Reference Number | Fix |
| 9552 | System Failure. Unable To Perform Customer Profile Request at This Time. | Call |
| 9553 | Invalid Action Indicator | Fix |
| 9555 | Invalid BIN | Fix |
| 9556 | Invalid Merchant ID | Fix |
| 9557 | Invalid Name | Fix |
| 9558 | Invalid Address | Fix |

| Code | Description | Action |
|---|---|---|
| 9559 | Invalid Address 2 | Fix |
| 9560 | Invalid City | Fix |
| 9561 | Invalid State | Fix |
| 9562 | Invalid ZIP | Fix |
| 9563 | Invalid Email | Fix |
| 9564 | Invalid Phone | Fix |
| 9565 | Invalid Order Description | Fix |
| 9566 | Invalid Amount | Fix |
| 9567 | Invalid Account Type Indicator | Fix |
| 9568 | Invalid Account Number | Fix |
| 9569 | Invalid Account Expire Date | Fix |
| 9570 | Invalid ECP Account DDA | Fix |
| 9571 | Invalid ECP Account Type Indicator | Fix |
| 9572 | Invalid ECP Account Route | Fix |
| 9573 | Invalid ECP Bank Payment Delivery Method | Fix |
| 9574 | Invalid Switch Solo Start Date | Fix |
| 9575 | Invalid Switch Solo Issue Number | Fix |
| 9576 | Unable to Perform Profile Transaction. The Associated Transaction Failed. | Call |
| 9577 | Invalid Order Override Indicator | Fix |
| 9578 | Merchant-Bin combination is not allowed to perform profile transactions. | Call |
| 9579 | Merchant-Bin is not active. | Call |
| 9580 | Cannot process profile for Cust Ref Num and MID combination. A database error has occurred | Call |
| 9581 | Cannot process profile. Profile does not exist for Cust Ref Num and MID. | Fix |
| 9582 | Cannot process profile. Profile already exists for Cust Ref Num and MID. | Fix |
| 9583 | Missing Switch Solo Account Information. Either start date or issue number is required. | Fix |
| 9584 | Missing Electronic Check Account Information. | Fix |
| 9585 | Missing Credit Card Account Information. | Fix |
| 9587 | Auto-Gen Cust Ref Num Error. | Call |
| 9588 | Unable to Determine Profile Action from Auth Request | Fix |
| 9589 | Cannot Create Profile: A Customer Profile Name is Required | Fix |
| | **Retry Errors** | |
| 9710 | Message expired during retry | Resend |
| 9711 | Too many transactions to process | Wait & Resend |
| 9712 | Request timeout - Please try again | Resend |
| 9713 | Invalid MIME header - Merchant ID in MIME does not match XML message | Fix |
| 9714 | Invalid MIME header- Trace number must be between 1 and 9999999999999999 | Fix |
| 9715 | The retry request did not match the original request for this trace number | Fix |

| Code | Description | Action |
|---|---|---|
| | **IP Authentication Errors** | |
| 9716 | Security Information is Missing | Call Customer Service |
| 9717 | Security Information - agent/chain/merchant is missing | Call Customer Service |
| | **Web Server Errors** | |
| 20400 | Invalid Request | Fix |
| 20403 | Forbidden: SSL Connection Required | Fix |
| 20408 | Request Timed Out | Resend |
| 20412 | Precondition Failed: Security Information is missing | Call Customer Service |
| 20500 | Internal Server Error | Resend |
| 20502 | Connection Error | Resend |
| 20503 | Server Unavailable: Please Try Again Later | Resend |

# ProfileProcStatus Response Values

| Code | Definition | Action |
|---|---|---|
| 0 | Profile Action Successful | None |
| 9550 | Invalid Customer Reference Number From Order Indicator | Fix |
| 9551 | Invalid Customer Reference Number | Fix |
| 9552 | System Failure. Unable To Perform Customer Profile Request at This Time. | Call |
| 9553 | Invalid Action Indicator | Fix |
| 9555 | Invalid BIN | Fix |
| 9556 | Invalid Merchant ID | Fix |
| 9557 | Invalid Name | Fix |
| 9558 | Invalid Address | Fix |
| 9559 | Invalid Address 2 | Fix |
| 9560 | Invalid City | Fix |
| 9561 | Invalid State | Fix |
| 9562 | Invalid ZIP | Fix |
| 9563 | Invalid Email | Fix |
| 9564 | Invalid Phone | Fix |
| 9565 | Invalid Order Description | Fix |
| 9566 | Invalid Amount | Fix |
| 9567 | Invalid Account Type Indicator | Fix |
| 9568 | Invalid Account Number | Fix |
| 9569 | Invalid Account Expire Date | Fix |
| 9570 | Invalid ECP Account DDA | Fix |
| 9571 | Invalid ECP Account Type Indicator | Fix |
| 9572 | Invalid ECP Account Route | Fix |

| Code | Definition | Action |
|---|---|---|
| 9573 | Invalid ECP Bank Payment Delivery Method | Fix |
| 9574 | Invalid Switch Solo Start Date | Fix |
| 9575 | Invalid Switch Solo Issue Number | Fix |
| 9576 | Unable to Perform Profile Transaction. The Associated Transaction Failed. | Call |
| 9577 | Invalid Order Override Indicator | Fix |
| 9578 | Merchant-Bin combination is not allowed to perform profile transactions. | Call |
| 9579 | Merchant-Bin is not active. | Call |
| 9580 | Cannot process profile for Cust Ref Num and MID combination. A database error has occurred | Call |
| 9581 | Cannot process profile. Profile does not exist for Cust Ref Num and MID. | Fix |
| 9582 | Cannot process profile. Profile already exists for Cust Ref Num and MID. | Fix |
| 9583 | Missing Switch Solo Account Information. Either start date or issue number is required. | Fix |
| 9584 | Missing Electronic Check Account Information. | Fix |
| 9585 | Missing Credit Card Account Information. | Fix |
| 9587 | Auto-Gen Cust Ref Num Error. | Call |
| 9588 | Unable to Determine Profile Action from Auth Request | Fix |
| 9589 | Cannot Create Profile: A Customer Profile Name is Required | Fix |

# CurrencyCode and CurrencyExponent Tag Values

| Currency | Code | Exponent |
|---|---|---|
| Albanian Lek | 008 | 2 |
| Algerian Dinar | 012 | 2 |
| Angolan New Kwanza | 982 | 2 |
| Argentine Peso | 032 | 2 |
| Armenian Dram | 051 | 2 |
| Aruban Guilder | 533 | 2 |
| Azerbaijanian Manat | 031 | 2 |
| Bahamian Dollar | 044 | 2 |
| Bangladeshi Taka | 050 | 2 |
| Barbados Dollar | 052 | 2 |
| Belarussian Ruble | 974 | 0 |
| Belize Dollar | 084 | 2 |
| Bermudian Dollar | 060 | 2 |
| Bolivian Boliviano | 068 | 2 |
| Bosnia Convertible Mark | 977 | 2 |
| Botswana Pula | 072 | 2 |
| Brazilian Real | 986 | 2 |
| Brunei Dollar | 096 | 2 |
| Bulgarian Lev | 975 | 2 |

| Currency | Code | Exponent |
|---|---|---|
| Burundi Franc | 108 | 0 |
| Cambodian Riel | 116 | 2 |
| Cape Verdi Escudo | 132 | 2 |
| Cayman Islands Dollar | 136 | 2 |
| Cfa Franc Bceao | 952 | 0 |
| Cfa Franc Beac | 950 | 0 |
| Cfp Franc | 953 | 0 |
| Chilean Peso | 152 | 2 |
| Chinese Yuan Renminbi | 156 | 2 |
| Colombian Peso | 170 | 2 |
| Comoro Franc | 174 | 0 |
| Costa Rican Colon | 188 | 2 |
| Croatian Kuna | 191 | 2 |
| Cyprus Pound | 196 | 2 |
| Czech Koruna | 203 | 2 |
| Djibouti Franc | 262 | 0 |
| Dominican Peso | 214 | 2 |
| East Caribbean Dollar | 951 | 2 |
| Ecuador Sucre | 218 | 2 |
| Egyptian Pound | 818 | 2 |
| El Salvador Colon | 222 | 2 |
| Eritrea Nakfa | 232 | 2 |
| Estonian Kroon | 233 | 2 |
| Ethiopian Birr | 230 | 2 |
| Falkland Islands Pound | 238 | 2 |
| Fiji Dollar | 242 | 2 |
| Gambian Dalasi | 270 | 2 |
| Georgian Lari | 981 | 2 |
| Ghanaian Cedi | 288 | 2 |
| Gibraltar Pound | 292 | 2 |
| Guatemala Quetzal | 320 | 2 |
| Guinea Franc | 324 | 2 |
| Guinea-Bissau Peso | 624 | 2 |
| Guyanese Dollar | 328 | 2 |
| Haitian Gourde | 332 | 2 |
| Honduras Limpera | 340 | 2 |
| Hungarian Forint | 348 | 2 |
| Iceland Krona | 352 | 2 |
| Indian Rupee | 356 | 2 |
| Indonesian Rupiah | 360 | 2 |
| Iranian Rial | 364 | 2 |
| Iraqi Dinar | 368 | 2 |
| Israeli New Shekel | 376 | 2 |
| Jamaican Dollar | 388 | 2 |
| Kazakhstan Tenge | 398 | 2 |

| Currency | Code | Exponent |
|---|---|---|
| Kenyan Shilling | 404 | 2 |
| Kyrgyzstan Som | 417 | 2 |
| Laos Kip | 418 | 0 |
| Latvian Lats | 428 | 2 |
| Lebanese Pound | 422 | 2 |
| Liberian Dollar | 430 | 2 |
| Lithuanian Litas | 440 | 2 |
| Macau Pataca | 446 | 2 |
| Macedonia Denar | 807 | 2 |
| Malagasy Franc | 450 | 0 |
| Malawi Kwacha | 454 | 2 |
| Malaysian Ringgit | 458 | 2 |
| Maldive Rufiyaa | 462 | 2 |
| Maltese Lira | 470 | 2 |
| Mauritania Ouguiya | 478 | 2 |
| Mauritius Rupee | 480 | 2 |
| Mexican Peso | 484 | 2 |
| Moldovan Leu | 498 | 2 |
| Mongolia Tugrik | 496 | 2 |
| Moroccan Dirham | 504 | 2 |
| Mozambique Metical | 508 | 2 |
| Myanmar Kyat | 104 | 2 |
| Namibia Dollar | 516 | 2 |
| Nepalese Rupee | 524 | 2 |
| Netherlands Antillean Guilder | 532 | 2 |
| New Guinea Kina | 598 | 2 |
| Nicaraguan Cordoba Oro | 558 | 2 |
| Nigerian Naira | 566 | 2 |
| North Korean Won | 408 | 2 |
| Pakistan Rupee | 586 | 2 |
| Panamanian Balboa | 590 | 2 |
| Paraguay Guarani | 600 | 0 |
| Peruvian Nuevo Sol | 604 | 2 |
| Philippines Peso | 608 | 2 |
| Polish Zloty | 985 | 2 |
| Qatari Rial | 634 | 2 |
| Romania Leu | 642 | 2 |
| Russian Ruble | 643 | 2 |
| Rwanda Franc | 646 | 0 |
| Samoan Tala | 882 | 2 |
| Sao Tome & Principe Dobra | 678 | 2 |
| Saudi Riyal | 682 | 2 |
| Seychelles Rupee | 690 | 2 |
| Sierra Leonean Leone | 694 | 2 |
| Slovak Koruna | 703 | 2 |

| Currency | Code | Exponent |
|---|---|---|
| Slovenian Tolar | 705 | 2 |
| Solomon Islands Dollar | 090 | 2 |
| Somali Shilling | 706 | 2 |
| South Korean Won | 410 | 0 |
| Sri Lanka Rupee | 144 | 2 |
| St. Helena Pound | 654 | 2 |
| Sudanese Dinar | 736 | 2 |
| Swaziland Lilangeni | 748 | 2 |
| Syrian Pound | 760 | 2 |
| Taiwan Dollar (New) | 901 | 2 |
| Tanzanian Shilling | 834 | 2 |
| Thai Baht | 764 | 2 |
| Tonga Pa'anga | 776 | 2 |
| Trinidad & Tobago Dollar | 780 | 2 |
| Turkish Lira (New) | 949 | 2 |
| Turkmenistan Manat | 795 | 2 |
| Uganda Shilling | 800 | 0 |
| Ukrainian Hryvnia | 980 | 2 |
| United Arab Emirates Dirham | 784 | 2 |
| Uruguayan Peso | 858 | 2 |
| Uzbekistan Sum | 860 | 2 |
| Vanuatu Vatu | 548 | 0 |
| Venezuelan Bolivar | 862 | 2 |
| Vietnamese Dong | 704 | 2 |
| Yemeni Rial | 886 | 2 |
| Yugoslavian Dinar (New) | 891 | 2 |
| Zambian Kwacha | 894 | 2 |
| Zimbabwe Kwacha Dollar | 716 | 2 |

# Purchasing Card 3 Tables

## ISO Country Codes

| ISO Code | Country |
|---|---|
| GBR | UNITED KINGDOM |
| DEU | GERMANY |
| FRA | FRANCE |
| AUT | AUSTRIA |
| BEL | BELGIUM |

| ISO Code | Country |
|---|---|
| CAN | CANADA |
| CZE | CZECH REPUBLIC |
| DNK | DENMARK |
| FIN | FINLAND |
| GRC | GREECE |
| HUN | HUNGARY |
| ISL | ICELAND |
| IRL | IRELAND |
| ISR | ISRAEL |
| ITA | ITALY |
| LIE | LIECHTENSTEIN |
| LUX | LUXEMBOURG |
| MEX | MEXICO |
| NLD | NETHERLANDS |
| NOR | NORWAY |
| POL | POLAND |
| SVK | SLOVAKIA |
| ESP | SPAIN |
| SWE | SWEDEN |
| CHE | SWITZERLAND |
| USA | UNITED STATES |
| ARE | UNITED ARAB EMIRATES |
| AUD | AUSTRALIA |
| HKD | HONGKONG |
| JPY | JAPAN |
| MYR | MALAYSIA |
| NZD | NEW ZEALAND |
| SGD | SINGAPORE |
| ZAD | SOUTH AFRICA |

## Unit of Measure

| UOM Code | Unit Name |
|---|---|
| ACR | Acre |
| ASM | Alcoholic strength by mass |
| ASV | Alcoholic strength by volume |

| UOM Code | Unit Name |
|---|---|
| AMP | Ampere |
| AMH | Ampere-hour (3,6 kC) |
| ARE | Are (100 m2) |
| BAR | Bar |
| BLL | Barrel (petroleum) (158,987 dm3) |
| BFT | Board foot |
| BQL | Becquerel |
| BIL | Billion EUR |
| MLD | Billion US |
| BHP | Brake horse power (245,7 watts) |
| BTU | British thermal unit (1,055 kilojoules) |
| BUA | Bushel (35,2391 dm3) |
| BUI | Bushel (36,36874 dm3) |
| CDL | Candela |
| CCT | Carrying capacity in metric tonnes |
| CNT | Cental GB (45,359237 kg) |
| CGM | Centigram |
| CLT | Centilitre |
| CMT | Centimetre |
| DTN | Centner, metric (100 kg) |
| WCD | Cord (3,63 m3) |
| COU | Coulomb |
| CKG | Coulomb per kilogram |
| CMQ | Cubic centimeter |
| DMQ | Cubic decimeter |
| FTQ | Cubit foot |
| INQ | Cubic inch |
| MTQ | Cubic metre |
| MQH | Cubic metre per hour |
| MQS | Cubic metre per second |
| MMQ | Cubic millimetre |
| YDQ | Cubic yard |
| CUR | Curie |
| DAY | Day |
| DAA | Decare |
| DLT | Decilitre |
| DMT | Decimetre |
| DTN | Decitonne |
| CEL | Degree Celsius |
| FAH | Degree Fahrenheit |
| | Degree Kelvin: see Kelvin |
| DPT | Displacement tonnage |
| DZN | Dozen |
| DZP | Dozen packs |
| DZR | Dozen pairs |
| DCP | Dozen pieces |
| DRL | Dozen rolls |
| DRM | Drachm GB (3,887935 g) |
| DRI | Dram GB (1,771745 g) |
| DRA | Dram US (3,887935 g) |
| BLD | Dry barrel (115,627 dm3) |
| GLD | Dry gallon (4,404884 dm3) |
| PTD | Dry pint (0,55061 dm3) |

| UOM Code | Unit Name |
|---|---|
| QTD | Dry quart (1,101221 dm3) |
| FAR | Farad |
| OZI | Fluid ounce (28,413 cm3) |
| OZA | Fluid ounce (29,5735 cm3) |
| FOT | Foot (0,3048 m) |
| GLI | Gallon (4,546092 dm3) |
| GBQ | Gigabecquerel |
| GWH | Gigawatt-hour (1 million kW/h) |
| GII | Gill (0,142065 dm3) |
| GIA | Gill (11,8294 cm3) |
| GRN | Grain GB, US (64,798910 mg) |
| GRM | Gram |
| GFI | Gram of fissile isotopes |
| GGR | Great gross (12 gross) |
| GRO | Gross |
| GRT | Gross (register) ton |
| SAN | Half year (six months) |
| HAR | Hectare |
| HBA | Hectobar |
| HGM | Hectogram |
| DTH | Hectokilogram |
| HLT | Hectolitre |
| HPA | Hectolitre of pure alcohol |
| HMT | Hectometre |
| HTZ | Hertz |
| HUR | Hour |
| CEN | Hundred |
| BHX | Hundred boxes |
| HIU | Hundred international units |
| CLF | Hundred leaves |
| CNP | Hundred packs |
| CWA | Hundredweight US (45,3592 kg) |
| INH | Inch (25,4 mm) |
| JOU | Joule |
| KEL | Kelvin |
| KBA | Kilobar |
| KGM | Kilogram |
| KPH | Kilogram of caustic potash |
| KSH | Kilogram of caustic soda |
| KNS | Kilogram of named substance |
| KNI | Kilogram of nitrogen |
| KPP | Kilogram of phosphonic anhydride |
| KPP | Kilogram of phosphorus pentoxide |
| KPH | Kilogram of potassium hydroxide |
| KPO | Kilogram of potassium oxide |
| KSH | Kilogram of sodium hydroxide |
| KSD | Kilogram of substance 90% dry |
| KUR | Kilogram of uranium |
| KMQ | Kilogram per cubic meter |
| KGS | Kilogram per second |
| KHZ | Kilohertz |
| KJO | Kilojoule |
| KMT | Kilometre |

| UOM Code | Unit Name |
|---|---|
| KMH | Kilometre per hour |
| KPA | Kilopascal |
| KTN | Kilotonne |
| KVR | Kilovar |
| KVT | Kilovolt |
| KVA | Kilovolt-ampere |
| KWT | Kilowatt |
| KWH | Kilowatt-hour |
| KNT | Knot (1 nautical mile per hour) |
| LEF | Leaf |
| GLL | Liquid gallon (3,78541 dm3) |
| PTL | Liquid pint (0,473176 dm3) |
| QTL | Liquid quart (0,946353 dm3) |
| LTR | Litre (1dm3) |
| LPA | Litre of pure alcohol |
| CWI | (Long) hundredweight GB (50,802345 kg) |
| LTN | Long ton GB, US (1,0160469 t) |
| LUM | Lumen |
| LUX | Lux |
| MHZ | Megahertz |
| MAL | Megalitre |
| MAM | Megametre |
| MPA | Megapascal |
| MVA | Megavolt-ampere (1000 KVA) |
| MAW | Megawatt |
| MWH | Megawatt-hour (100 kW/h) |
| MTR | Metre |
| MTS | Metre per second |
| MSK | Metre per second squared |
| CTM | Metric carat (200 mg = 2.10-4 kg) |
| TNE | Metric ton (1000 kg) |
| MLD | Milliard |
| MBR | Millibar |
| MCU | Millicurie |
| MGM | Milligram |
| MLT | Millilitre |
| MMT | Millimetre |
| MIO | Million |
| HMQ | Million cubic metres |
| MIU | Million international units |
| MIN | Minute |
| MON | Month |
| NMI | Nautical mile (1852 m) |
| NTT | Net (register) ton |
| NEW | Newton |
| NMB | Number |
| NAR | Number of articles |
| NBB | Number of bobbins |
| NCL | Number of cells |
| NIU | Number of international units |
| NMP | Number of packs |
| NMR | Number of pairs |
| NPL | Number of parcels |

| UOM Code | Unit Name |
|---|---|
| NPT | Number of parts |
| NRL | Number of rolls |
| OHM | Ohm |
| ONZ | Ounce GB, US (28,349523 g) |
| APZ | Ounce GB, US (31,10348 g) |
| PAL | Pascal |
| DWT | Pennyweight GB, US (1,555174 g) |
| PCE | Piece |
| PTI | Pint (0,568262 dm3) |
| LBR | Pound GB, US (0,45359237 kg) |
| PGL | Proof gallon |
| QTI | Quart |
| QAN | Quarter (of a year) |
| QTR | Quarter, GB (12,700586 kg) |
| DTN | Quintal, metric (100 kg) |
| RPM | Revolution per minute |
| RPS | Revolution per second |
| SCO | Score |
| SCR | Scruple GB, US (1,295982 g) |
| SEC | Second |
| SET | Set |
| SHT | Shipping ton |
| SST | Short standard |
| STN | Short ton GB, US (0,90718474 t) |
| SIE | Siemans |
| CMK | Square centimeter |
| DMK | Square decimeter |
| FTK | Square foot |
| INK | Square inch |
| KMK | Square kilometer |
| MTK | Square metre |
| MIK | Square mile |
| MMK | Square millimeter |
| TDK | Square yard |
| WSD | Standard |
| ATM | Standard atmosphere (101325 Pa) |
| SMI | (Statute) mile (1609,344 m) |
| STI | Stone GB (6,350293 kg) |
| ATT | Technical atmosphere (98066,5 Pa) |
| DAD | Ten days |
| TPR | Ten pairs |
| MIL | Thousand |
| TAH | Thousand ampere-hour |
| MBF | Thousand board feet (2,36 m3) |
| TQD | Thousand cubic metres per day |
| MBE | Thousand standard brick equivalent |
| TSH | Ton of steam per hour |
| TNE | Tonne (1000 kg) |
| TSD | Tonne of substance 90% dry |
| TRL | Trillion EUR |
| BIL | Trillion US |
| APZ | Troy Ounce |
| LBT | Troy pound, US (373,242 g) |

| UOM Code | Unit Name |
|---|---|
| VLT | Volt |
| WTT | Watt |
| WHR | Watt-hour |
| WEB | Weber |
| WEE | Week |
| YRD | Yard |
| ANN | Year |

# *Appendix B*

## General Card Validation

There are three common edits that catch the greatest majority of bad card numbers:

- MOD 10 check digit
- Credit card prefix check
- Credit card length validation

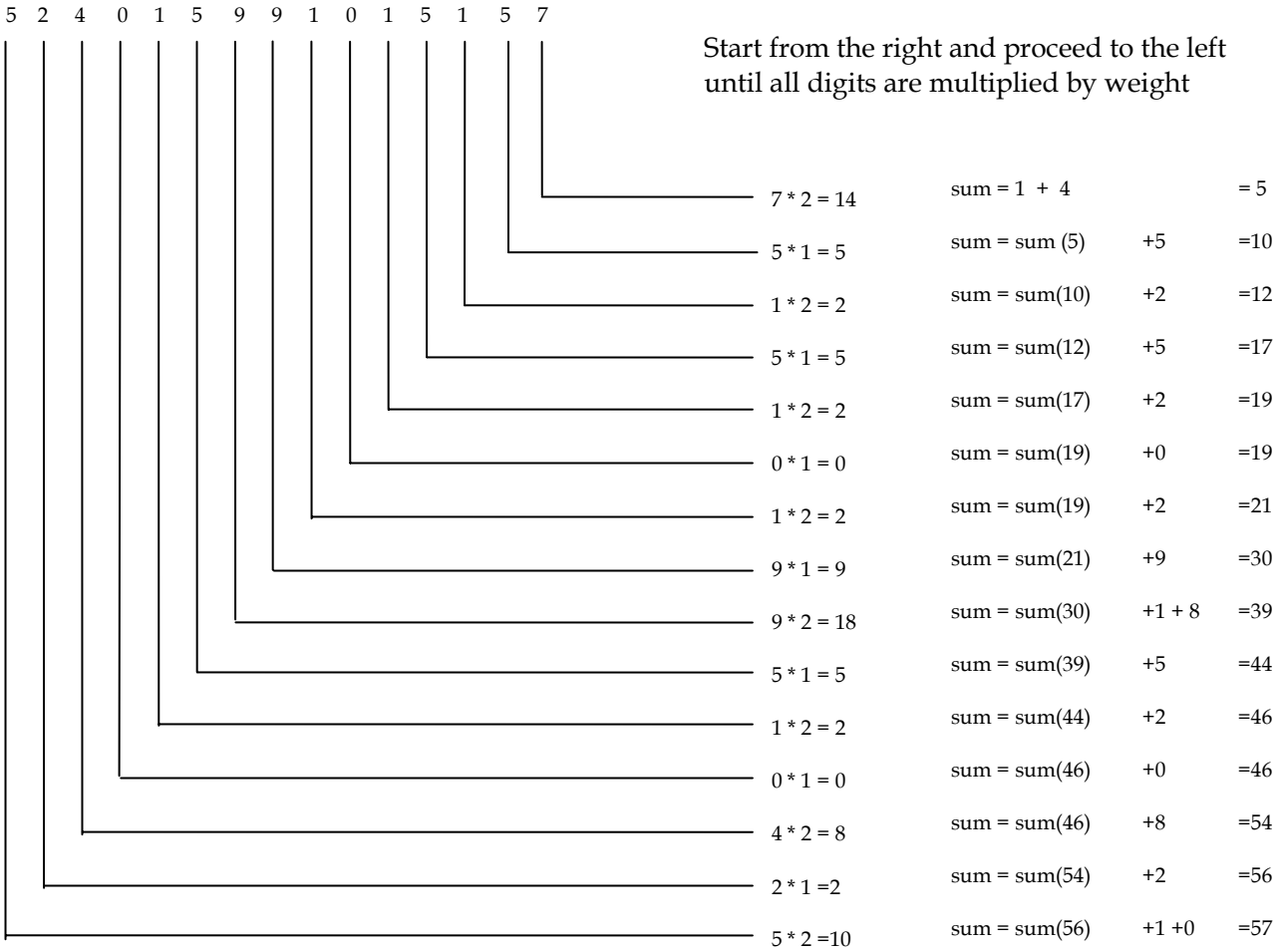
### MOD 10 Check Digit

The MOD 10 check digit calculation validates the credit card by calculating the last digit of the card number from all the other numbers in the card.

The last digit of a credit card can be calculated based on a calculation performed upon all the digits preceding it. This operation is called a **MOD 10 check-digit routine.**

*Continued on next page*

Use the following card for Example: 5240159910151573


5 2 4 0 1 5 9 9 1 0 1 5 1 5 7
Start from the right and proceed to the left until all digits are multiplied by weight
7 * 2 = 14
5 * 1 = 5
1 * 2 = 2
5 * 1 = 5
1 * 2 = 2
0 * 1 = 0
1 * 2 = 2
9 * 1 = 9
9 * 2 = 18
5 * 1 = 5
1 * 2 = 2
0 * 1 = 0
4 * 2 = 8
2 * 1 =2
5 * 2 =10
sum = 1 + 4 = 5
sum = sum (5) +5 =10
sum = sum(10) +2 =12
sum = sum(12) +5 =17
sum = sum(17) +2 =19
sum = sum(19) +0 =19
sum = sum(19) +2 =21
sum = sum(21) +9 =30
sum = sum(30) +1 + 8 =39
sum = sum(39) +5 =44
sum = sum(44) +2 =46
sum = sum(46) +0 =46
sum = sum(46) +8 =54
sum = sum(54) +2 =56
sum = sum(56) +1 +0 =57

Remove the **check digit**, 3, which is present in this example card.
sum = 57
sum MOD 10 ———> 57 MOD 10 = 7
10-7 = **3**
check digit of 5240159910151573 is **3**.

*Continued on next page*

### Example Coding for MOD 10

The following routine is a check digit routine written in the 'C' programming language:

```
/* The operator for mod in 'C' is % */
long mod10 (card,card_len-1)          /* module 10 check digit function */
char *card;                           /* credit card number */
short card_len                        /* card length */
{
register int count;                   /* a counter */
register int weight;                  /* weight to apply to digit being checked*/
register int sum;                     /* sum of weights */
register int digit;                   /* digit being checked*/
long    mod;

weight=2;
sum     =0;

 /*       compute the sum */
for (count = card_len - 1; count >=0; count = count -1)
        {
        digit = weight * (card[count] - '0');

        /* add both the tens digit and the ones digit to the sum */
        sum = sum + (digit / 10) + (digit % 10);

        if (weight ==2)
                weight = 1;
        else
                weight = 2;
        }
/*  subtract the ones digit of the sum from 10 and return the ones digit of that
result */
mod = (10 - sum%10) % 10;
return (mod)
}
```

## Card Prefix Check

The prefix check is the comparison of the first few digits of each card number to a list of known prefixes.

| Card Type | Prefix |
|---|---|
| American Express / Optima | 37, 34 |
| Bill Me Later | 504990, 621993 |
| Carte Blanche | 389 |
| Diners Club | 30, 36, 381 - 388 |
| Discover (Novus) | 60110, 60112, 60113, 60114, 60119 |
| JCB | 3528 – 3589 |
| MasterCard | 51 – 55 |
| PINLess Debit | See PINLess Debit Section |
| Switch/Solo [BIN 000001 ONLY] | 49, 56, 6* where * is any single digit |
| Visa/Delta | 4 |

## Card Length Check

The number of digits for each card is constant, allowing a validation to be performed by verifying the number of digits for each card number.

| Card Type | Length |
|---|---|
| American Express / Optima | 15 |
| Bill Me Later | 16 |
| Carte Blanche | 14 |
| Diners Club | 14 |
| Discover (Novus) | 16 |
| JCB | 16 |
| MasterCard | 16 |
| PINLess Debit | 12 – 19 |
| Switch/Solo [BIN 000001 ONLY] | 16, 18, or 19 |
| Visa/Delta | 13 or 16 |

**End of Document**